



**PEGA –
DEV 8.7v**

Table of Contents

1. SAMPLE APPLICATION	- 3 -
1.1 Class Group or Work Pool	- 5 -
1.2 Studios - PEGA	- 7 -
1.3 Operator ID & Access Group	- 8 -
1.4 Case Creation	- 10 -
2. SAMPLE WEBPAGE	- 11 -
2.1 Conditional Layouts and Properties	- 12 -
2.1 Unit Testing	- 13 -
2.2 Clipboard	- 14 -
2.3 Activities	- 16 -
2.4 Data Transform	- 20 -
2.4 Data Tables	- 24 -
2.5 Advanced Controls	- 25 -
3. OTHER RULES	- 27 -
3.1 Flow Action	- 27 -
3.2 Flows-Types	- 28 -
3.3 Validations	- 30 -
3.4 Decision Rules	- 32 -
3.5 Circumstance	- 36 -
3.6 Routing	- 37 -
3.7 SLA (Service level Agreement)	- 39 -
3.8 Advance Shapes (Parallel Processing)	- 43 -
3.9 Declarative Rules	- 44 -
3.10 Tracer	- 46 -
4. RULESET, INHERITANCE	- 48 -
4.1 Ruleset & Ruleset Version	- 48 -
4.2 Rule Availability	- 49 -
4.3 Application Inheritance	- 54 -
4.4 Application Validation/ Ruleset Validation	- 56 -
4.5 Access Group:	- 57 -

4.6 Product File or Package or RAP File	- 63 -
4.7 Delegation	- 67 -
5. INTEGRATIONS	- 68 -
5.1 Service-SOAP	- 69 -
5.2 Connect - SOAP.....	- 73 -
5.3 How to handle the Exceptions in Integrations	- 75 -
5.4 Service REST.....	- 76 -
5.5 Connect REST	- 79 -
5.6 Dynamic System Settings(DSS)	- 81 -
6. DATA PAGES (DECLARATIVE PAGES)	- 83 -
6.1 Types of Data Pages:	- 88 -
6.2 Types of Data Access Patterns in PEGA:	- 90 -
7. REPORT DEFINITION	- 91 -
7.1 Types of Reports.....	- 91 -
7.2 How to Expose or Optimize the Properties in PEGA	- 94 -
7.3 How to Join the two or More than two data base tables in pega	- 96 -
7.4 How to Schedule the Reports to Business Users in PEGA.....	- 98 -
7.5 How to Connect with External Data base Table in PEGA.....	- 101 -
8. LOOPING & OBJ- METHODS	- 106 -
8.1 How to Generate PDF File	- 117 -
9. LISTENERS	- 119 -
9.1 EMAIL Listeners	- 119 -
9.2 FILE Listeners	- 121 -
9.3 Agents	- 123 -
9.4 Queue Processor	- 127 -
10. CASE MANAGEMENT	- 129 -
10.1 Skin	- 136 -

1. SAMPLE APPLICATION

administrator@pega.com → User Name

install → Password

Pega Platform -----→ Base Application

PEGA OOTB rules, Functions, Methods → Reuse → Custom Application

Application Name: - Bank, Retail, Booking, Complaint, Insurance,

Organization Name: - SBI, HSBC, BigBazar, REDBus, Amazon, LIC, Wipro

Division Name: - SBH, SBM India USA JAPAN Chennai, Hyd Bangalore,

Unit Name: - Marketing, HRDept, FinanceDept, HealthInsurance,
VehicalInsurance

1. Implementation Level Layer

Particular Module related rules or Specific to the implementation application related rules we will create under Implementation Level Layer

2. Framework level Layer

In a Framework level Layer, we will create Implementation application related Reusable rules or common rules or reusable Implementation application functionalities or components

3. Class Group

Class Group is a combination of Abstract classes and Concrete classes

Implementation Level Class → Organization Name - Division Name - Unit Name -
Application Name - Work

Framework level Class → Organization Name – FW - Application Name - Work

SBI-FW-Bank-Work

Insurance FW APP

Pega Platform → Insurance FW App → LIC FW App → LIC Health Insurance Imp
App

Pega Platform → Insurance FW App → LIC FW App → LIC Vehicle Insurance Imp
App

1.1 Class Group or Work Pool

Class Group is a combination of abstract classes and Concrete classes. It contains different category rules.

1. Organization Name - Division Name - Unit Name - Application Name - Work

Ex: SBI-Hyderabad-Sale-BankAp-Work → Implementation Class

2. Organization Name – FW - Application Name – Work (*FW APP*)

Ex: SBI – FW – BankAp - Work → Framework level Class

Note: In General Class is a nothing but Object, it will combine the variables and methods

Note: In Pega Point of View, Class is nothing but combination different category rules like "Sections, Flow Actions, Flows, Data Transform, Activities...etc"

Abstract Class:

1. In a Abstract class level, we cannot create Work Objects
2. In a Abstract class level, we cannot do any implementation means we can not write any business logic.

Concrete Class

1. In a Concrete class level, we can create Work Objects.

In a Concrete class level, we can do implementation means we can write business logic.

CLASS STRUCTURE:

In my Class Structure contains 5 Layers

1. **Organization Level Layer:-** In a Organization Level layer we will create all the divisions related commons rule, Frameworks related common rules, Company Logos, Security Related rules.

Ex:SBI

2. **Division Level Layer:-** In a Division Level Layer we will create all the units related common rules and specific division related rules.

Ex:SBI-Hyderabad SBI-Bangalore SBI-Chennai

3. **Unit Level Layer:-** Particular unit related rules we will create under unit level layer

Ex: SBI-Hyderabad-Sale SBI-Hyderabad-HRDept SBI-Hyderabad-Finance

4. **Implementation Level Layer:-** Particular module related rules or Specific Implementation Application related rules we will create under Implementation Level Layer

Ex: SBI-Hyderabad-Sale-BankAp-Work-CreditCard CreditCard Num

SBI-Hyderabad-Sale-BankAp-Work-Claim Claim ID

SBI-Hyderabad-Sale-BankAp-Work-Insurance Insurance Id

5. **Framework Level Layer:-** In a Framework level layer we will create common rules or reusable rules or reusable functionalities and all the implementation application related rules.

Ex: SBI-FW-BankAp-Work → Framework Level Class

1.2 Studios - PEGA

PEGA 8

Dev Studio

Admin Studio

App Studio

Prediction Studio

PEGA 7

Designer Studio

SMA(System Management Application)

1.3 Operator ID & Access Group

How to Create Operator IDs in PEGA

kishore -----> Operator ID

http://localhost1:8080/prweb/PRServlet/app/default/ → PEGA Development Env Cloud URL

http://localhost2:8080/prweb/PRServlet/app/default/ → PEGA SIT Env Cloud URL or Staging1

http://localhost3:8080/prweb/PRServlet/app/default/ → PEGA UAT Env Cloud URL or Staging2

http://localhost4:8080/prweb/PRServlet/app/default/ → PEGA PreProd Env Cloud URL

http://localhost5:8080/prweb/PRServlet/app/default/ → PEGA Production Env Cloud URL Prod Env

Goto the Records Explore → Create → Organization → Operator ID

Access Group

Access group will control the Access of the Application to specific Group or Users (Operators)

In Pega 8.7 V:

BankAp:Authors ---->Developer

BankAp:Users

Pega 8.3 Below versions

BankAp:Administrators -->Developer

BankAp:Users

BankAp:Managers

BankAp:Authors

<https://www.icicibank.com/>

Developers

Customers

Employees

Managers

SrManagers

HRs

Marketing

Directors

CEO

Clone --->Model Operator ID

1.4 Case Creation

Example:

LOAN: Collect Documents Verification Approval Money Dispatched

CREDITCARD: Documents Verification Approval/Reject Card Dispatched

1. Define a "Case Type"

Note: "Case Type" is a nothing but Module.

Ex: HomeLoan

2. We need to define the "Stages" to complete the particular Case Type.

Ex: *Document Collection, Document Verification, Manager Approval* → Stages

3. We need to define the "Steps" to complete the particular Stage

Ex: *PersonalInfo, EmployeeInfo, SalaryInfo* → Steps

4. We need to configure the form to each and every step

Note: Configure the form is a nothing but add a Properties (Fields or variable name)

Note: All the CaseTypes are inherited from "Work-" class. Whatever classes are inherited from "Work-" class, all the classes "belongs to class group".

Ex:SBI-Hyderabad-Sale-BankAp-Work-HomeLoan

SBI-Hyderabad-Sale-BankAp-Work-Order

2. SAMPLE WEBPAGE

1. Create a Properties and Specify the Property type & Control

Right Click on Class → Create → Data Model → Property

Note:

- ➔ If we want to display the Dropdown property in a UI then we need to select under UIControl Dropdown to "pxDropDown"
- ➔ If we want to display the CheckBox property in a UI then we need to select under Property type to "TrueFalse"

2. Create a Section and Drag & drop the Properties into the Section

Note: Section is a nothing but "User Interface (UI)", We will arrange the properties into the section some proper format means how we want to show UI to the End Users.

Right Click on Class → Create → User Interface → Section

3. Create Flow Action and Call the Section into the Flow Action

Right click on class → Create → Process → Flow Action

4. Create a Flow and call the Flow action in a Flow under "Outgoing connector" of each assignment.

Right click on class → Create → Process → Flow

Note: By using **SubProcess** shape in a flow we can call from one flow to another flow

2.1 Conditional Layouts and Properties

1. Create When rule based upon Source Property

Right Click on class → Create → Decision → When rule

2. Call the When rule in a Section under Target Property or Target Layout
3. Refresh the Section under Source Property

.VehicleName = 'HERO Motercorp' || .VehicleName = 'TVS'

Ex1: Gender → Male / Female

Are You Married → Yes / No (Radiobutton)

Spouse Name: Husband Name :

Ex2: Do you Credit Card → Check box

Card Type: Card limit: Credit Card Bank Name:

Ex3: Are You Working Employee → Yes / No (Dropdown)

Company Name : Years of Exp: Technology: Company Location

2.1 Unit Testing

From Dev Studio → Launch Web Interface → Click on User Portal

➔ In Real time point of view we need to click or select customized portal

User Story: UTR Document (Unit Test Results like Positive and Negative Scenarios)

LIVE UI : "LIVE UI" is a one of the Debugging tool, By Using "LIVE UI" we can identify the "Properties, Section and Flow Actions" at the run time.

Field Value: For Longer TextBox, CheckBox, Dropdown, RadioButton captions or reusable label messages we will create Field Value

Right Click on Class → Create → Data Model → Field Name

pyCaption → TextBox, CheckBox, Dropdown, RadioButton

pyButtonLabel → Button Captions

pyCategoryLavel → Error Message Captions

Binary File: For Images, Excels, PDF, Video files purpose we will create Binary File

Goto the Records Explore → Create → Technical → Binary File

2.2 Clipboard

1. Clipboard is a one of the Debugging tool, whatever we are entering the data in a UI we can see the data in a clipboard under "User Pages" with page name "pyWorkPage".
2. Clipboard is a temporary memory means it will not save the data into the clipboard permanently.

PEGA UI<-----Through Clipboard----->Data Base Table

PEGA App<-----Through Clipboard----->External Application

Note:

1. If we want to see any work object related data in a clipboard then we should open the work object then only we can view the data in a clipboard
2. When we are submitting the assignment, Pega itself will commit the data into "Work" Data base Table.

1. UserPages

pyWorkPage --->it will pointing to Work object class name

newAssignPage --->it contains assignment related data

pyAttachmentPage -->it contains the attachment related data

Custom Pages (whatever developers are going to define in a activity level based upon the requirement)

2.DataPages

1.Thread

2.Requestor

3.Node

3. Linked Property Pages

4. System Pages

Application

OperatorID

AccessGroup

Organization

Division

pxRequestor

2.3 Activities

Whenever we want to do the data base operations like Insert, Update, Delete, Open, Browse or If we want to transfer the data from PEGA Application to External application or vice versa or If we want to insert the data into Excel, CSV, Txt files and place the files into Application Server Path or If we want to read the file from Application Server path and insert or update the data into the data base table or if we want to track the log messages like success or failure scenarios then we can go for "Activities"

Right Click on class → Create → Technical → Activity

PEGA UI<-----Insert,Update,Delete,Open,Browse----->Data base Table

PEGA App<-----Data Transfer----->External Application or System (JAVA /Don Net)

PEGA App<-----Excel,CSV,Txt----->Application Server(Tomcat or Weblogic or WebSphere)

PEGA App<----Success or Failure Senarios-----Log Messages

1. Parameter Variable:

Parameter variable values we can pass from one rule to another rule

→ By Using "Param" keyword we can access the Parameter variable values

Ex: Country → Param.Country

2. Local Variable:

Local Variable values we can access within the rules means we can't pass from one rule to another rule

→ By using "Local" Keyword we can access the Local Variable values

Ex: State

Local.State

3. Pages & Classes

Pages & Classes is a just like references, If we want to access one class rules into another class then we can specify the page name and class Name. By Using Referenace Page name we can access the Reference class rules into the current class.

4. Activity type

Activity type will define "How we can call the activity into some other rules "

Note: If Activity type is "Utility" then we can call the activity in a flow by using "Utility" shape.

5. Set the Values to class Properties

Property Name: Property Value

.FirstName: Chandra

.LastName: Sekhar

.Age: 26

.Location: Vizag

Property-Set → By Using Property-Set Method in a activity we can set the values to class properties.

Page-New → By Using "Page-New" method in a activity we can create new Page in Clipboard.

1.RunRecordPrimaryPage

2.pyWorkPage

3.SamplePage (Custom Page)

6. How to Call from one Activity to Another Activity

Call: By Using "Call" we can call from one Activity to another activity.After Completion of the Sub Activity execution it will come to the current activity and will execute remaining methods after "Call" Method.

Branch: By Using "Branch" we can call from one Activity to another activity.After Completion of the Sub Activity execution it will end the current activity and sub activity execution means it will not execute any methods "Branch".

Queue: By Using "Queue" we can call from one Activity to another activity.If we use the "Queue" method then it will execute current activity and sub activity parallely

Note: We cannot see the Parameter Variable values and local variable values in a Clipboard.

Note: We can see the Parameter Variable values and local variable values in a "Tracer".

7. How to Pass parameters from one Activity to another Activity

If we want pass the parameters from one Activity to another Activity then we need to check the "Pass Current Parameter Page" check box under "Call or Branch" methods in an activity. Before "Call or Branch" methods we need to set the Parameters by using Property-Set methods.

How to See the Parameter Variable Values in a Tracer

➔ In a Tracer, we need to click on the particular Step method and it will open one window, there we can see the "Parameter Page name and Unnamed". Now we need to click on "UnNamed" to see the parameter variable values in a Tracer.

How to see the Local variable values in a Tracer

- ➔ In a Tracer we need to check the "Local Variable" check box under Settings Tab. We need to click on that particular step method in a tracer and it will open one window, There we can see the "Local variable and View Variable". Now we need to click on "ViewVariable" to see the Local Variable Values in a Tracer.

How to Call or When to Call the Activity in a Section, Flow Action and Flow

- ➔ If we want to call the Data Transform or Activity or Some other rules while changing the text box property value or Clicking the Radiobutton or checking check box or changing the Dropdown value or Clicking the Buttons in a UI then we need to goto the section and call the the Data Transform or Activity or Some other rules in that particular property level.
- ➔ If we want to call the Data Transform or Activity while loading the screen then we need to call the activity or data transform in a flow action under "Action" Tab we have "Pre Processing" option is available,there we can call the activity or data tranform.
- ➔ If we want to call the Data Transform or Activity while Submitting the screen then we need to call the activity or data transform in a flow action under "Action" Tab we have "Post Processing" option is available,there we can call teh activity or data tranform.

Property-Set-Messages

- ➔ If we want to set the error messages in a Property level/ Page Level then we can use the Property-Set-Messages method in a activity

2.4 Data Transform

Whenever we want to set the data to class properties or based upon when conditions if we want to set the values to class Properties or If we want to copy the data from One Page Property to another Page Property or One Page Pagelist property to another pagelist property or One Page Property to another pageList Property or If we want to sort the PageList property results in the form ascending or descending order or If we want to loop the List PageList Results then we can go for Data Transform.

- By Using Apply-Datatransform method in a activity we can call the data transform
- By Using "pxExecuteAnActivity" PEGA OOTB Function we can call the Activity in a data transform
- By Using "ObtainValue" PEGA OOTB Function we can call the Decision table in a data transform
- By using "pxCallRetrieveReportData" pega ootb activity we can call the report defination in a data transform.

Right Click on class → Create → DataModel → Data Transform

1. Set

If we want to set the values to class properties then we can select the "Set" Action

Property-Set → If we want to set the values to class properties then we can use the Property-Set Method in Activity

2. Remove

If we want to remove the Property or Page or pagelist Property values from Clipboard then we can use the "Remove" Action

Property-Remove → if we want to remove the Property value from the clipboard then we can use the Property-Remove method in a activity

Page-Remove → if we want to remove the Page from the clipboard then we can use the Page-Remove method in a activity

3. UpdatePage

If we want to copy the one Page Property data to another page property data then we can use the UPdatePage Action.

Page-Copy → if we want to copy the one Page Property data to another page property data then we can use the Page-Copy method in a activity.

Ex: CustomerPage: Customer ID Customer Name Customer Age Location

Selected Customer Page: Customer ID Customer Name Customer Age Location

4. Apply-DataTransform

If we want to call one data transform to another data transform then we can use the "Apply-DataTransform" action

5. Sort

If we want to sort the PageList property results in the form assending or decending order then we can use the "Sort" Action

Obj-Sort → If we want to sort the PageList property results in the form assending or decending order then we can use the "Obj-Sort" method in a activity (Ex:CustomerList 100)

6. Comment

If we want to provide the useage of data transform or step actions then we can provide the comments for future reference.

7. When, Otherwise When, Otherwise

If we want to check the Different combination of Property conditions then we can use the "When,Otherwise When ,Otherwsie" action

8. Append to

- ➔ If we want to copy the data from One Page Pagelist property to another pagelist property or One Page Property to another pageList Property then we can use the "Append to" Action
- ➔ When we use the "Append to", Source and Target pages should be same class.
- ➔ When we use the "Append to", at a time it will copy the soruce page or pagelist property values to target pagelist property.

Ex1: CustomerList (CustomerID CustomerName CustomerAge Location)

CustomerPage (CustomerID CustomerName CustomerAge Location)

9. Append and Map to

If we want to copy the data from One Page Pagelist property to another pagelist property or One Page Property to another pageList Property then we can use the "Append and Map to"

- ➔ When we use the "Append and Map to", Source and Target pages should not be a same class means different classes or same class also we can copy the data from Target pagelist to source page list or Source Page property to target pagelist property.
- ➔ When we use the "Append and Map to", it will copy the all source page or pagelist property values to target pagelist property or Specific Values.

Ex1: CustomerList (CustomerID CustomerName CustomerAge Location)

CustomerPage (CustomerID CustomerName CustomerAge Location)

10. For Each Page In

If we want to loop the list of pagelist results then we can use the "For Each Page In" action.

Ex:CustomerList

11. Exit For Each

If we want to exist from loop then we can use the "Exit For Each" action

12. Exit DataTransform

If we want to exit the data transform based when conditions or while looping then we can use the exit data transform

Q): *what is the use of "Call superclass data transform" check box in a data transform?*

In Case Same data transform is available under Parent class and child classes then if we check the "Call superclass data transform" check box in a data transform, first pega will call parent class data transform and then it will call child class data transform

Note:

1. By Using "Page or PageList or PageGroup" properties we can access one class properties into the Another class
2. substring, Trim, WhatComesAfterFirst, WhatComesAfterLast, WhatComesBeforeFirst, WhatComesBeforeLast

3. IndexInPagelist()

@IndexInPageList(true,".pySelected",pyWorkPage.DependentList)

4. pxExecuteAnActivity(Primary, pxAddMessageToProperty)

2.4 Data Tables

1. Create Datatype, Create Properties (Columns)
2. Select at least one property as a Primary Key
3. Insert the Data into Data base table

Note:

- All the Data types are inhirated from "Data-" class.Whatever classes are inhirated from "Data-" class, All the classes by default it will take "does not belongs class Group"
- In Pega point of view, Data Type is a nothing Data table.
- Data Type we can use as a class or data table.

Q): What are sources available for Table Layout" in a section?

1. Property: We need to provide PageList or PageGroup

2. Report Defination: We need to provide the Report defination name and class name

3.Data Page: We need to provide the List Data Page.

Log-Message:

By Using Log-Message method in a activity we can set the Log Messages like success or failure scenario's.

Debug —> Record as DEBUG severity, produces the largest number of messages; least severe.

Info —> Record as INFO severity.

Warn —> Record as WARN severity.

Error —> Record as ERROR severity; most severe.

InfoForced —> Always record as INFO event

2.5 Advanced Controls

1. Dropdown (DynamicSelect)

If we want to display the Values into Dropdown property from data base table dynamically then we can use the "DropDown" Control.

Sources:- Report Definition, DataPages, ClipboardPage (PageList or PageGroup Property)

2. CaseCade-Dynamic Select

Based Upon First Dropdown Property Values, Automatically it will display the values into second dropdown Property dynamically from data base table. this process we can call as CaseCade DynamicSelect.

Sources:- Report Definition, DataPages, ClipboardPage (PageList or PageGroup Property)

3. AutoComplete

Based upon minimum search characters If we want to display the values into the Droddown property from data base table dynamically then we can use "AutoComplete".

Sources:- Report Definition, DataPages, ClipboardPage (PageList or PageGroup Property)

4. RadioButtons

If we want to set the values to radiobutton properties dynamically from data base table then we can use the RadioButton control

Sources:- Report Defination, DataPages, ClipboardPage (PageList or PageGroup Property)

Property for value	:AP Andhra Pradesh	AP	Andhra Pradesh
Property for display text	:AP	AP	Andhra Pradesh Andhra Pradesh

5. MultiSelect

If we want to select the Multiple values from dropdown property then we can use the MultiSelect Control .

Sources:- Report Defination, DataPages, ClipboardPage (PageList or PageGroup Property)

6. Slidar

If we want to give the Rating then we can use Slidar control

3. OTRHER RULES

3.1 Flow Action

1. Connector Flow Action

Normally whatever we are creating a flow actions, those flow actions we can call as a Connector Flow Action

Connector Flow Action can be called in a flow under "Outgoing connector" of each assignment shape.

In a Connector Flow Action, work object will moves from one assignment shape to another assignment shape.

1--FA1--2---FA2--3--FA3---4

2. Local Flow Action

If we want to display the popups when we are clicking the "Buttons or Checkboxes or Radio buttons" then we can go for Local Flow Actions

Local Flow Action can be called in a flow under Assignment shape and sections under "Buttons or Checkboxes or Radio buttons".

In a Local Flow Action, Work Object will be in the same assignment means after completion of the local action work object will not move to next assignment shape.

Note:

- a. Flow wide local action can be called in a flow under "Design" tab we have "Flow-wide local actions" option is available, there we can call the local actions.
- b. Assignmnet level flow action can be called in a flow under assignment shape

- c. Case Wide Local action can be called in a "pyDefault" Case Type rule under "Process" tab we have "Case wide actions" option is available,there we can call the local actions.
- d. Stage Wide Local action can be called in a "pyDefault" Case Type rule under "Stage" tab we have "Optional actions" option is available,there we can call the local actions.

TICKETS:

By Using Tickets we can jump from one Assignment to another assignment

- a. Create Ticket rule
- b. Create Activity and call the Ticket rule in a Activity by using "SetTicket" PEGA OOTB Activity
- c. Call the Activity and ticket rule in a flow

Ex1: Approval Flow

1---2----3----4

As new Requirement I don't want to execute 2 and 3 Assignments

1---Call the Activity ---2-----3---4 under 4th Assign-shape call the Ticket rule

Ex2: Registration Process Flow

1-Personal Info---2--Verification--3----4---5 ReVerification Required Button

3.2 Flows-Types

1. Process Flow

- a) Normally whatever we are creating a flows, those flows we can call as process flow. In a Process flow we cannot jump from one screen to another screen or we cannot go back to the previous screen.

- b) By Using Process Flow we cannot achieve the Front & Back Navigation, Tabbed navigation and Tree Navigation
- c) In Process flow We can call the flow action under Outgoing connector of each assignment.
- d) In a Process flow, Individual assignment **Routing, SLA, Tickets & Local Actions** is available
- e) In a Process flow, Multiple users can work on entire process flow.
- f) In a Process flow we can call the process flow within the process flow and screen flow
- g) In a Process flow we can see the all-advanced shapes like "Split-Join, Split-For-Each, Integrator...etc"

2. Screen Flow

- a) Whenever User is going to fill the form, If user want to come back to the previous screen or jump from one screen to another screen then we can go for Screen flow
- b) By using Screen flow we can achieve the Front & Back Navigation, Tabbed navigation and Tree Navigation
- c) In A Screen flow we can call the Flow Action under Assignment shape
- d) In a Screen flow, Individual assignment routing, SLA, Tickets and local actions is not available but we can route entire screen flow under starter shape.
- e) In a Screen flow, Only one user can work on entire screen flow.
- f) In a Screen flow we can call screen flow within the screen flow but we can not call the process flow in a screen flow.

g) In a screen flow we can see the only one advanced shape like split for Each.

- Performed Screen Flow
- Tabbed Screen Flow
- Tree Navigation Screen Flow

Note:

- In a Screen flow Under "Starter" shape we can specify the Whether it is "Perform screen flow or tabbed screen flow or Tree Navigation Screen flow".
- If we check the "Save on last step" check box in a screen flow under starter shape then it will commit the entire screen flow data while submitting the last assignment.

3. Sub Flow

Sub flow is a nothing but calling a flow within the flow. By using "SubProcess" shape we can call from one flow to another flow.

For Reusable purpose we can create sub flows or In a Main flow if we want to reduce shapes then we will create sub flows.

3.3 Validations

If we want to validate whether user is entering correct data or not then we will create validation rules.

Note: Whatever validations we are doing in a Browser level, those validations we can call as a Client-Side validations. Whatever validations we are doing in a Server Level, those validation we can call as Server-Side Validations

TYPES OF VALIDATIONS

1. Edit Validation

- By using Edit Validation rule we can validate single Property.
- In a Edit Validation rule we need to write a java source code.
- Edit Validation rule can be called in a property level under advanced Tab
- By using "Property-Validate" method in a activity we can call the edit validation
- By Using Edit Validation rule we can achieve the Client Side Validations
- Edit Validation rule can be also called as Property Validation.
- If we want to do complex validations then we can go for Edit Validations

Email-Validation:

```
String A = "^[\\w-_.+]*[\\w-_.]@[\\w]+\\.([\\w]+[\\w])$";
boolean B=theValue.matches(A);

if(B==true)
{
    return true;
}
else
{
    theProperty.addMessage("Please Enter Valid Email ID");
    return false;
}
```

2. Obj Validation(Validate rule)

- By using Obj-Validation rule we can validate the multiple properties.
- Obj validation rule can be called in a flow action under Validation Tab
- By using Obj-validate method in a activity we can call the validation rule
- By Using Obj Validation rule we can achieve the Server Side Validations.

→ Obj Validation rule can also called as Page Validation

→ For Simple Validations we can go for Obj Validation rule

3. Edit Input

- a) If we want to convert the data from one format to another format then we can go for Edit Input
- b) For Edit Input rule we need to write a java source code.
- c) Edit Input rule rule can be called in a property level under advanced tab.

Ex:toUpperCase,toLowerCase

3.4 Decision Rules

Based upon Property Conditions if we want to return the values then we can go for Decision rules.

TYPES OF DECISION RULES

1. Decision Table

- a) If we want to implement "If Elseif Elseif ElseIf...etc" logic then we can go for Decision Table
- b) If First condition is true then it will return the value otherwise it will one by one condition and then return value.
- c) By using "Property-Map-DecisionTable" method in a activity we can call the decision table
- d) By Using "Decision" Shape in a flow we can call the Decision Table

- e) By Using "ObtainValue" pega ootb function we can call the decision table in a data transform
- f) Decision table can be called in a Decision Tree
- g) We can call the Decision table within decision table

`@DecisionTable.ObtainValue(tools, Primary, "SetTicketPrice")`

```
Ex: If(A<10)
    {
        Return X
    }
elseif(A>10 & A<20)
    {
        Return Y
    }
Elseif(A>20)
    {
        Return Z }
```

2. Decision Tree

- a) If we want to implement "If if if If...etc or nested If" logic then we can go for Decision Tree
- b) Whether First condition is true or false, it will check the all the conditions and then return the value.
- c) By using "Property-Map-DecisionTree" method in activity we can call the decision tree
- d) By Using "Decision" Shape in a flow we can call the Decision Tree
- e) Decision tree can not be called in a Decision Table
- f) We can call the Decision tree within decision tree
- g) Call in Data Transform by using `pxEvaluateDecisioinTree`

```

Ex: If(A<10)
{
    Return a
}
if(A>10 & A<20)
{
    Return b
}
if(A>20)
{
    Return c
}

```

```

If(A<10)
{
    If(B<15)
    {
        If(C<20)
        {
            Return K
        }
    }
}

```

3. Map value

- a) If we want to check multiple property conditions and return the multiple values then we can go for Map value
- b) In a Map value we will write logic in the "Matrix" Form
- c) By Using "Decision" shape in a flow we can call the Map value
- d) By Using "Property-Map-Value" method in a activity we can call the Map Value

1 1 1

1 1 1

1 1 1

4. When rule

- a) When rule will return either "True or False". If Condition is satisfied then it will return "True" otherwise it will return "False".
- b) When rule can be called in a Flow, Flow Action, Data Transform, Activities, Section...etc.

Q): What is the use of AllowMissingProperties check box in Decision Map Decision Table method?

If selected, the system attempts to evaluate the rule even when one or more properties referenced in the rule are not present on the clipboard.

Ex1: Type of Employment:Salary,Doctor,Business

Screen1 ---->Decision--->Screen2

Ex2: Bus Type:AC Non AC

Ticket Price:1200 700

Q): How to Call Decision Table in a Decision Table

In Decision Tree under Configuration Tab we need to check the "Allow selection of 'call decision' option" check box.

Q): When Decision Table will behave like Decision Tree

If we check the "Evaluate all rows" check box in decision table under Results tab then decision table will behave like decision tree

Q): Is it possible to call the Decision table within the Decision table?

By Using "ObtainValue" Pega ootb function we can call decision table within the decision table

3.5 Circumstance

Circumstance is a nothing but "Creating new rule based upon base rule". If Property conditions are satisfied then it will execute the circumstance rule otherwise it will execute base rule

Note: For Properties, File Listeners, Service-Pkgs, Connectors we cannot create circumstance rules.

TYPES OF CIRCUMSTANCES

1. **Single Property Circumstance:-** Based Upon single property condition we can create single property circumstance.
2. **Metaproperty or Multivariant circumstance:-** Based upon Multiple Property conditions we can create metaproperty circumstance

For Multiproperty circumstance we need to create below rules

1.Circumstance Template: In a Circumstance Template rule we need to add the Properties to execute multiproperty circumstance

2.Circumstance Defination: In a Circumstance Defination rule we need to add the Property conditions to execute the Multiproperty conditions.

3. **Date Property Circumstance:-** Based Upon Date Property condition or In between Date Conditions we can create Circumstance rule.

3.6 Routing

By Using Routing We can transfer Assignment or Work Object from one user (Operator) to Another User or Multiple Users(Operators). Routing Can be done in a flow under Assignment Shape.

TYPES OF ROUTING

1. WorkList(Operator)

By Using WorkList we can transfer the Assignment from one user to another user.

2. WorkQueue(Work Basket)

By Using WorkQueue we can transfer the Assignment from one user to Multiple users.

WorkBasket is a nothing but collection of work Items (Work Objects). Whatever work items are available under Work Queue, Multiple users can able to access.

Work Group/Team:- Work Group is a nothing but "Group of users working under Same Department or team"

Ex: VerificationDept WB

LoanApprovalDept WB

CardDispatchedDept WB

Ex1:- Leave Request Case Type

1--Leave Details--2--Review Leave Details and Approve--3 Send confirmation Email

Ex2:- Service Request Case Type

1-CSR user-----2-ServiceRequest WB-----3

Ex3: Claim Amount

Claim Amount < 100000 Manager

Claim Amount > 100000 < 200000 Senior Manager

Claim Amount > 200000 < 300000 Director

Dynamic Routing:-

→ Create Activity

↳ Property-set

↓
param.AssignTo

.OperatorName.

Security Tab.

Activity Type : Route

[Get operator Names
in Case ?
(UI)]

↓ Dropdown

Rep det

→ Open Assignment

Activity

Route to : Custom

Assignment type : Worklist

→ [ToWorklist]

Security Tab.

Activity Type : Route

Activity [Internal]

↳ check for substitute ()

↳ check availability

D. Table.

Route to : Custom

Assignment type : Worklist

Router : To Decision Table.

Give D. Table Name

☒ check availability

toWorklist

[in Searchbar]
@ Flow?

3.7 SLA (Service level Agreement)

SLA will define the "Time and urgency" to complete the particular task or assignment

SLA Can be called in a Flow under Assignment Shape

TYPES OF TIME INTERVALS

7pm Work Object Created Date Time July 10th

1.Goal Time:- 2hr 3days

Whenever Work Object reaches to that particular assignment shape, From that time onwards it will calculate the Goal Time.

After 2hr what is my goal time: $7\text{pm} + 2\text{hr} = 9\text{pm}$

2.Dead Line Time:- 3hr 5days

Whenever Work Object reaches to that particular assignment shape, From that time onwards it will calculate the Dead Line Time.

After 3hr What is my dead line time: $7\text{PM} + 3\text{hr} = 10\text{pm}$

Note: Dead line time always greater than Goal Time

3.Passed DeadLine Time:- 1hr 2days

From Dead line time completion time onwards, it will calculate the Passed Dead Line time.

After 1hr what is my passed deadline time: $10\text{pm} + 1\text{hr} = 11\text{pm}$

TYPES OF URGENCY (0-100)

- Default Urgency

Default Urgency Value is :10

- Initial Urgency: 20
- Goal Urgency: 15

After Completion of the Goal time, what is the urgency value:

Default+Initial+Goal

$$10+20+15=45$$

- Dead line Urgency: 25

After Completion of the Dead line time, what is the urgency value:

Default+Initial+Goal+DeadLine

$$10+20+15+25=70$$

- Pass Dead line Urgency: 10

After Completion of the Passed Dead line time, what is the urgency value:

Default+Initial+Goal+DeadLine+Passed DeadLine

$$10+20+15+25+10=80$$

TYPES OF SLAS

1. Assignment SLA: -Assignment SLA can be called in a flow under Assignment shape
2. Process Level SLA (Flow Level): -Process Level SLA can be called in a flow under "Process Tab"
3. Stage Level SLA: -Stage Level SLA can be called in a Case type under Stage Level

4. Case Level SLA: -CASE Level SLA can be called in a Case Type under Settings Tab

HOW TO TRACE THE SLA

➔ From Admin Studio we can trace the SLA

1. Goto the Admin Studio and Select Resources Tab
2. Click on Agents, Now we can see the list agents. Which agent we want to trace we need to select that particular agent(Pega Procom Agent Ruleset ,ServiceLevelEvents Agent Name)
3. Click on Tracer

[Admin → Resources → Agents]

→ Dynamic SLA

↳ create Date-Time property

↓
refer it in Goal-Time [when not available in Operator-ID]

↳ refers to another operator

↳ Calls Activity while running SLA

Mock

→ addn. Pending

↳ Create Local-Action.

Ex:- Addn verification.

↳ Resend option [Action Tab]

→ To WorkBasket

Agents → to close

SLA WorkBasket

System Queue
Service Level

↓
Table to store [class]

SLA {Queue Items}

pr-Sys-Queue-SLA

Table name

Agent name [Pega-Program]

runs @ 30secs...

Service Level Agent

Pega Pre-defined activities

pyAdjustSLA

↳ reassign SLA values

paAdjustSLA

↳ Reset SLA

Reassign / Reassign To WorkBasket

↳ Reassign work to someone object

workbasket to someone

Create activity

→ pyAdjustSLAWrapper

SLA, SLA Name

→ Assignment Related Data will be stored [pc-Assign-Worklist]

→ Workbasket [pc-Assign-WorkBasket]

Assign-WorkBasket class

Assign-Worklist class

3.8 Advance Shapes (Parallel Processing)

1. Spin Off

If we check the Spint Off check box under SubProcess shape then main flow and sub flow will execute parallelly.

Ex: Approval Flow

1----2--Sub Process Shape----3

| If we check the Spint Off check box under SubProcess shape

Clearance Flow 1---2

2. Split Join

By using Split Join advanced shape we can call the multiple subflows in a main flow.

If we use the Split Join advanced shape in a main flow then any of the sub flow Execution completed or All the Sub flows Execution completed or Some of the Sub flow execution completed then only it will resume the Main flow.

Ex: Approval Flow

1-----2--Split Join (All, Any, Some) --3-----4

^

/ \

1---2 1--2

APF1 APF2

3. Split For Each

Split For Each will allow to run the "Single Process Flow" multiple times by Looping the PageList Property.

Ex1: Claim Approval Flow

1---2--Split For Each (All, Any, Some, Iterate)--3

|
1--2 Claim Review Process Flow (Claim Managers 3)

Ex2: Order Case type

TV,Laptop,Washing,AC

WarrentSupport SubCase Type

Note:- In Case Type we can add the "Parallel Process"

3.9 Declarative Rules

Declarative Rules can be created based upon the Properties. Wherever we are referring the properties, automatically it will call declarative rules means declarative rules no need to call separately.

TYPES OF DECLARATIVE RULES

1. Declare Expression

When we are changing the Source Properties values(Input),Automatically it will update the target property values(OutPut).this process we can call as Declare Expression. (DT, Activity, Decision Table)

Forward Chaining:- When we are changing the Source Properties values(Input),Automatically it will update the target property values(OutPut). this process we can call as Forward chaining

Ex: $A=B+C$

SubToal=Product Price * Product Qty

BackWard Chaining:- Wherever we are referring the Target Property value, Automatically it will recalculate the Source Property values.

Ex:A=B+C

Total=SubTotal+GST

Q): Where we can specify the Whether it is forward or backward chaining?

In a Declare Expression under Change Tracking tab we can specify the Whether it is forward or backward chaining.

Q) When we can say Forward Chaining?

Whenever Input Changes

Q) When we can say BackWard Chaining?

1.Wherever we are refering Property

2.Property Missing

2. Declare Constraints

By Using Declare Contraints we can validate the Multiple property values. Declare Constraints No need to call separately. By using Declare Constraints we can achieve the Client-Side Validations

Ex:Screen1 Screen2 Screen3

Age Age Age

3. Declare On Change

Whenever we are changing the Property Values, automatically it will run activity. this process we can call as Declare On Change

- a. Create When rule
- b. Create Activity
- c. Create Declare On Change and call the Call the When rule and activity

4. Declare Trigger

Whenever we are updating or Deleting or Saving or Commit the Instances of the Classs, automatically it will run activity. this process we can call as Declare Trigger

5. Declare Index

6. Data Pages(Declarative Page)

3.10 Tracer

Tracer is a one of the debugging tools. By using Tracer we can identify the what are the rules running in the back group while running the application or We can identify the Exceptions or Errors or Product Live Issues.

Remote Tracing:

1. From Admin Studio, Click on Resources and Requestors
2. Goto the Active Requestors Tab and select the Specific Requestor ID
3. Click on Trace

Or

1. From Dev Studio,...

CLASS - 34 → TRACER [Live UI, Clipboard, Wheream]

→ which rules are calling [while running application]

→ Identify Exceptions / Errors! [1--2--3(x)--4]

→ DB Query running in the background.

→ Debug Live Issues. [Generally support Team] → Developer doesn't has this.

Developer's can't create work-objects in Prod. Env. [Except Blockers]

Stream rules → [sections & Flow Actions]

→ Settings Tab [For req only Tracing]
Max Trace Display [5000] → default

Break points :-

→ Total 30 steps [Trace upto 10 steps only] [Only Activities from Initial only]

Watch Variable :-

→ Page Type data (specific)

Remote Tracer :-

→ diff requestor rules running in other system / Browser

Lower environments

→ Dev / SIT / UAT / pre-prod

pySetField Defaults

• paOrgenyzhlostClass

??

Check log → replicate in lower env → remote tracing [by requestor ID] → Schedule Phone call ✓

Save → Product Issues [Functionalities which are not working]

py default case type { default values }

→ To change workobject pre-fix.

4. RULESET, INHERITANCE

4.1 Ruleset & Ruleset Version

Ruleset & Ruleset Version is a nothing but Collection of Rules. Whatever we are creating a rule, all the rules will be available under Ruleset & Ruleset Version. By using Ruleset & Ruleset Version we can achieve the Security and Version Control.

Note: -Same rule is available under multiple Rulesets versions, Pega at the run time always will call the higher version rule.

How to Create Ruleset version: -

How to Lock the Ruleset Version: -

Check OUT:- Whenever we want to modify any unlocked Ruleset version related rule then we need to "Check Out" rule and do the changes.

Check In:- Whatever we did the changes in our Check Out mode,if everything is working fine as expected then we need to "check In" rule.Once we "Check In" rule, every one can able to see whatever we did the changes.

Discard:- Whatever we did the changes in our "Check Out" or PrivateEdit mode If we dont want those changes then we can "Discard" Our changes.

SaveAs:-If we want to modify any locked version rule then we need to "SaveAs" to Open ruleset version.

Delete:- If we want to delete any rule then we can click on "Delete" and provide Comments.

PrivateEdit:- If we want to modify the Locked version rule then we need to click on "Privatedit" and do the Changes.

Requirement DocumentV1

Requirement DocumentV1.1

Requirement DocumentV1.2

Release1 → 3months

Sprint1 2weeks Sprint2 2weeks Sprint3 Sprint4

10 User Stories 15 User Stories

5 Developers

Major-Minor-Patch

Bank3:01-01-01 50 rules

CustomerDetails:First Name Last name Age

Bank3Int:01-01-01 15 rules

New Requirment

CustomerDetails:Additional Properties

Full Name Middle Name Address

Bank3:01-01-01,02

CustomerDetails

4.2 Rule Availability

Rule Availability will define whether we can use the rule or not at the run time.

TYPES OF AVAILABILITIES

Available: - If rule availability is "Available", That rule is available to particular ruleset and rule version

Ex: CustomerDetails:01-01-01,01-01-02,03,04,05

Not Available: - If rule availability is "Not Available", That rule is Not available to particular ruleset and rule version

Final: - If rule availability is "Final", we can't override or we can't change final rules.

Withdrawn: - If rule availability is "Withdrawn", That rule is not available to Current ruleset & rule version and Lower versions of Current Ruleset version. In case Same rule is available under Parent classes then it will call from Parent classes.

Ex: CustomerDetails:01-01-01,01-01-02,03,04,05
HDFCB-India-Sale-Bank3-Work-CreditCard

Ex: CustomerDetails:01-01-01,01-01-02,03,04,05
HDFCB-India-Sale-Bank3-Work-CreditCard

HDFCB-India-Sale-Bank3-Work

HDFCB-India-Sale-Bank3

CustomerDetails:01-01-01

HDFCB-India-Sale

HDFCB-India

HDFCB

Blocked:- If rule availability is "Blocked", That rule is not available to Current ruleset & rule version, Lower versions of Current Ruleset version and Inherited (Parent) classes lower versions also.

Ex: CustomerDetails:01-01-01,01-01-02,03,04,05

HDFCB-India-Sale-Bank3-Work-CreditCard

HDFCB-India-Sale-Bank3-Work

HDFCB-India-Sale-Bank3

CustomerDetails:01-01-01

HDFCB-India-Sale

HDFCB-India

HDFCB

Class Inheritance

Class Inheritance is a nothing Acquiring (calling) the rules from Parent class to child class

Parent class----Possible----->Child Class

Parent class<----Not Possible-----Child Class

TYPES OF CLASS INHERITANCE

1. Pattern Inheritance

HDFCB-India-Sale-Bank3-Work-CreditCard

CustomerDetails:Name Age Location

HDFCB-India-Sale-Bank3-Work

HDFCB-India-Sale-Bank3

HDFCB-India-Sale

CustomerDetails:Address State

HDFCB-India

HDFCB

2.Direct Inheritance

Framework level Class HDFCB-FW-Bank3-Work

Work-Cover-

Work-

@baseClass

Explanation

When we are running the Application, PEGA first it will check whether rules are available or not. If Rules are available in a Pattern Inheritance then it will call from Pattern Inheritance otherwise it will check with Direct Inheritance. If rules are available under Direct Inheritance then it will call from Direct Inheritance.

Pattern Inheritance first it will check with Class Group level classes. If rules are available under Class Group level classes then it will call from Class Group level classes otherwise it will check with all Parent classes(Inherited classe). If rules are available in parent classes then it will call from parent classes otherwise it will check with Direct Inheritance.

Direct Inheritance first it will check with whether rules are available in a Framework level class or not. if Rules are available under Framework level class then it will call from Framework Level class other wise it will check with Pega Provided predefined classes like "Work-Cover-,Work-,@baseclass".

HDFCB-India-Sale-Bank3-Work-CreditCard

HDFCB-India-Sale-Bank3-Work-CarLoan

HDFCB-India-Sale-Bank3-Work-PersonalLoan

Ex: CustomerDetails RajaSri Check out

Name Age Address Customer EMail ID Phone Number

CustomerDetails Raam PrivateEdit

Last name State Zip

SKIMMING

Skimming means it will copy the rules into Higher rule set version from existing higher ruleset version. (Or) Copy the highest (major or minor) version of existing rules to a higher version.

Configure--->System--->Refactor--->Rulesets

TYPES OF SKIMMING

1.Major Skimming

While Doing Major Skimming, it will copy "Available, Not Available, Final and Blocked" rules.

Note: Withdrawn rules will not carry forward while doing Major Skimming.

Bank1:02-01-98

Bank1:03-01-01 Major skimming

2.Minor Skimming

While Doing Minor Skimming, it will copy "Available, Withdrawn, Final and Blocked" rules.

Note: Not Available rules will not carry forward while doing Minor Skimming.

Bank1:01-03-67

Bank1:01-04-01 Minor Skimming

4.3 Application Inheritance

Application Inheritance is a nothing but Calling a from rules from Framework application to Implementation Application

Framework Application-----Possible--->Implementation Application

Framework Application-----Not Possible--->Implementation Application

Pega Platform--->Framework Application--->Implementation Application

Rule Authorization

Rule Authorization will define whether User is authorized or access the rules or not at the run time.

Operator ID--->Access Group-->Application--->List Of Rulesets -->List of Rules

Rule Resolution Algorithm

Rule Resolution Algorithm will pick the best rule at the run time.

Explanation

1ST WAY

1. Whenever we are running the Application, PEGA will check whether rules are available in a "Rule Cache" or not.
2. If Rules are available in a Rule Cache then Pega Will check whether rule availability is blocked or Not. What are the rules blocked it will discard those rules and process the remaining rules.

Ex:10 Rules 3 rules 7 rules

3. Now it will check whether user is authorized to use the rules or not at the run time. Whatever rules are not authorized it will discard those rules and process remaining best rules at the run time.

Ex:7 rules 2 rules 5 rules.

2ND WAY

1. If rules are not available in a Rule Cache then pega will find the list of matching rules while running the application
2. now it will check whether rules are following Application Hierarchy (Application Inheritance) or Not. Whatever rules are not following Application Hierarchy it will discard those rules and process the remaining rules.
3. It Will check whether rule availability is Not Available or Not.What are the rules Not Available it will discard those rules and process the remaining rules.
4. It Will check whether rule availability is WithDrawn or Not.What are the rules Withdrawn it will discard those rules and process the remaining rules.

5. Now it will check whether rules are following Class Hierarchy(Class Inheritance),Ruleset Hierarchy and circumstance or Not.Whatever rules are not following Class Hierarchy(Class Inheritance),Ruleset Hierarchy and circumstance it will discard those rules and process the remaining rules.
6. Now it will place the rules into Rule Cache
7. If Rules are available in a Rule Cache then Pega Will check whether rule availability is blocked or Not.What are the rules blocked it will discard those rules and process the remaining rules.

Ex:10 Rules 3 rules 7 rules

8. Now it will check whether user is authorized to use the rules or not at the run time.Whatever rules are not authorized it will discard those rules and process remaining best rules at the run time.

4.4 Application Validation/ Ruleset Validation

Application validation is a nothing "Calling a rules from Lower Application related rulesets version into the Higher Application Ruleset versions rule" Ruleset Validation means it will follow ruleset hierarchy to call the rules into the current Ruleset version rules.

PegaRules--->AIG Level1 Imp App---->AIG Level2 Imp App---->AIG Level3 Imp App

List of Rulesets List of Rulesets List of Rulesets

Application--->List of Rulesets--->List of Rulesets

How to Validation Application rules/ Rulesets rule

It identify the Errors in Application ruleset related rules or Specify ruleset version related rules.

From Dev Studio-->Click Configure-->Application-->Tool-->Validation

Types of Request Types

1.Browser(H): User can access the pega application through Browser, at that time pega will create "Browser Requestor Type".

2.Application(A): When External Application is going to request the data from Pega Application at that time pega will create "Application Requestor Type".

3.Batch(B): When Background Processing jobs like Agents, Listeners are running in pega application, at that time pega will create "Browser Requestor Type".

4.Asynchprocesssor(B): job scheduler or Queue Processor running will run on Asynchprocesssor request type in pega application

5.Portlet(P): Portlet web application will generate dynamic contents, when Portlet web application component connecting with Pega applciation, at that time Pega will create "Portlet Requestor Type"

4.5 Access Group:

1. Access Group will control the "Access of the Application" to particular User or Group.
2. We can add the Many number Access groups to Specific Operator but only one Application we can access at a time.

3. While Creating a Access Group we need to provide the below details.

- Provide the Application Name and version

Ex: Bank or Insurance or Bank1 or Bank2 or Retail or Complaint

- Add the Available Portals

Ex: user Portal or Dev Studio or Case Worker or Case Manager or App Studio or Admin studio

- Add the Access Roles

Ex: FinanceDept, CustomerSupport

- We need provide the Work Pool or Class Group Name.

Ex: BOI-Hyderabad-Sale-Bank5-Work

ICICI Application

Developers

Employees ICICI App

Customers ICICI APP

Managers ICICI APP

MarketingDept

HRDept

FinanceDept

ICICI Application

Imp App CreditCard Dev Team1

Imp App Loan Dev Team2

Imp App Insurance Dev Team3

While Creating New Application on top of the Theme-Cosmos Application, Pega by default, it will create two Access Groups.

1.Bank:Authors --->Developers

2.Bank:Users --->Users

While Creating New Application on top of the PEGARULES Application, Pega by default it will create four Access Groups.

1.Bank:Administrators -->Developers

2.Bank:Users

3.Bank:Managers

4:Bank:Authors

Access Role:

1. Access Role will control access of the particular functionalities to particular Operator or Group.
2. In a Access Role we will add the "Access Role of Object".

ICICI Application

CreditCard

InternetBanking

Loan

Insurance

Access Role of Object:

In a Access Role of Object we will provide the class name and access rating (0 to 5) to Control "Create the instances (Work Object) or Open the Instances or Delete the Instances or Create a rules or Open the rules or Delete rules"

Ex:BOI-Hyderabad-Sale-Bank5-Work-CreditCard

0 or Blank -->No access

1 --->Exterimental Testing Env

2 --->Dev Env

3 --->SIT env

4 --->UAT Env

5 --->Prod Env

10.2 Access Deny

1. Access Deny will restrict the Access
2. In Access Deny we will specify the class name and Access Rating (0 to 5) To "Delete the Instances or Open the instaces or Create a Instances or Open the rule or Delete the rule or Create a Rule".
3. Access Role of Object will provide the Access Rating

Access Rating:-

0 or Blank ---> we are providing the Access

1 -->No Access ---> Experimental Testing Env

2 -->No Access ---> Dev Env

3 -->No Access ---> SIT Env

4 -->No Access ---> UAT env

5 -->No Access ---> Prod Env

Note: Access Role of Object and Access Deny will control the Access in a "class Level"

Privilege:

Privilege will control the access in a "rule level" like Flows, Flow Action, Activities..etc

Goto Records Explorer-->Security-->Create-->Privilege

➔ Step1: Create Privilege rule

➔ Step2: Add the Privilege rule in a access role of Object and Rule level

or

➔ Step1: Create Privilege rule

➔ Step2: Create Seperate Access role and Access the Access of Object

➔ Step3: In a Access role of object we need to call the privileg rule

➔ Step4: Add the Access role in a Access Group

➔ Step5: add the Privileg rule in a Rule level

Branching

By using Branching, we can achieve the Parallel Development.

➔ Open the Application rule from Application Menu and click on Add Branch

Ex1: Onshore Team Branch ➔ Baji

Offshore Team Branch ➔ Chandra

Ex2: GetCustomerDetails

Developer1Branch

Developer2Branch

→ Branching :- ^{Create +} [Merge + Issues] → Manually check & resolve ✓
↳ for Parallel development, + Versions Merging
Ex:- Team 1 { Customer details sections.
Team 2

Every Project follows
Branching

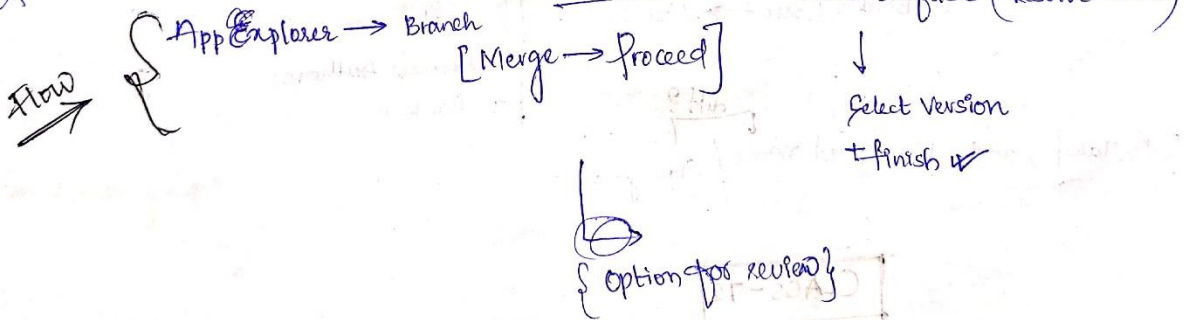
1. App def

↳ add branch

2. While selecting branch, It won't show ruleset version
{ Particular rule }

↳ Creates "Branch ruleset"
Branch > Ruleset

Flow



4.6 Product File or Package or RAP File

Before Creating a Product file, we need to follow below steps

- ➔ Whatever rules we want to move to other Environments we need "Check In" the all rules.
- ➔ Lock the ruleset versions whatever we want to move to Other Environments

Goto Records Explorer → Create → SysAdmin → Product File

- 1) We need to Include Application and Application version

Ex: Bank3:01-01-01

- 2) We need to Include Rulesets and Ruleset versions

Ex: Bank3:01-01-04

Bank3Int:01-01-07

HDFCB:01-01-03

- 3) We need to Include Class Instances

Ex: HDFC-Data-Policy 1000 Records

- 4) We need to include Individual instances ("pzInskey" of Non Ruleset version related rules).

Ex: Dev Env(Rules)---Move-->SIT Env---->UAT Env--->PreProd Env ---
>Production Env or Live Env

Q): How to Move the non-Ruleset version rules into SIT or UAT or Production Env in pega

Ans): we need to take the "pzInsKey" value of that particular rules and include in Product file under "Include Individual Instances"

Q): How to Move the new Data Base Table into Other Environments in Pega

- ➔ *Include "pzInsKey" value of "Data base Table class Name"*
- ➔ *Include "pzInsKey" value of "Data Base mapping rule"*

Q): How to Take the "pzInsKey" value of any rule in pega?

- ➔ *Open the Rule and Goto Action menu*
- ➔ *Select View XML of that particular*
- ➔ *Search for "pzInsKey"*

Q): How to Include the "Data Table Instances(Records)" in a Product file?

- ➔ *Inlcude the class name under "Include Class Instances"*

Ex: HDFC-Data-Order

Q) How to include Specific Record (Instance) of Any Data base table or Data Type in Product file?

We will include the pzInsKey valaue of Specific Record and include into Product file.

Q) How to Include Specific rule into the Product file ?

we need to take the "pzInsKey" value of that particular rule and include in Product file under "Include Individual Instances"

Ex:Bank3Prod:01-01-03 SetCustomerDetails DT

Sprint1

Bank3:01-01-04: CustomerDetails 05

Bank3Int:01-01-07

HDFCB:01-01-03

Access Group

WB

WG

DSS

File Listener

Production Ruleset Bank3Prod:01-01-02

New Data base Table Creation

Note: We will Prepare the "Release Note Document". While Deploying we will the follow the "Release Note Document".

Pre Deployment Tasks:

1. If any Back up required we need to take the back up of any Data base table
2. In case If we want to Stop the Agents or Listeners or Job Scheduler, we will do this activity

HOW TO DEPLOY PRODUCT FILE INTO OTHER ENVIRONMENTS

1. we need to Login into Specific Environment
2. From Dev Studio, Click on "Configure menu → Application → Destruction → Import"
3. Select Product file from your Local System and click on Next → Next → Finish

Post Deployment Tasks:

1. We need to Verify Ruleset Count
2. If Any DSS value or Data base table Records to update
3. Any Data Pages Flush
4. Any Agents or Listeners or Job Schedulers If we want to restart then we will this activity from Admin Studio
5. We need to Do the Regression Testing
6. From Business (Client) side will do Regression testing

Note:In my Project we will do the Production Deployment every 2weeks

Sprint Duration:2weeks 3 weeks

Production Deployment Process:

1. We will create one "ChangeRequest" in a ServiceNow(Incidents management) Tool.
2. We will Send email about Product Deployment details to Product owner and Business Ppl.

Ex: Deployment Date:August 26th

Time: Non Business Working Hours from 9PM EST to 11PM EST

No Business Impact

Sprint3 Changes

3. Again, we will send the Email confirmation about deployment completion to Product Owners and Business.

4.7 Delegation

Delegating a rule is a nothing but "modifying the rules in LIVE Env"

1.Dicision Table/Decision Tree

2.Correspondence

3.Data Types

4.SLA

- A. Create Product Ruleset and Version
- B. Add the Production Ruleset and Version in Application Definition rule
- C. Add the Product Ruleset and version in a Access Group level
- D. Whatever rule we want to delegate, Open that rule and click on Actions menu
- E. Select Delegate and provide the Access Group Name and Description

Note: In Manager Portal Under "Configuration" Tab we can see the list of Delegated Rules.

5. INTEGRATIONS

Connect-SOAP

Service-SOAP

Connect-REST

Service-REST

Connect-File

Service-File

Service-Email

Connect-SQL

Application1(PEGA) <-----Data Transfer----->Application2(JAVA)

Application1(PEGA) <-----Web Service (SOAP,REST)----->Application2(JAVA)

Connectors

- a. In a Connectors, PEGA Application is going to give the request. Based upon request, External Application or Source will send the response to Pega Application.
- b. In a Connectors, External Application or Source will provide the WSDL URL or File.

WSDLURL:

Pega Applciation-----Request----->External Application/Source/System

Pega Applciation<-----Response-----External Application/Source/System

Services

- a. In a Services, External Application is going to give the request. Based upon request, Pega Application will send the response to External Application/Source/System.
- b. In a Services, PEGA Application will provide the WSDL URL or File.

Pega Applciation<-----Request-----External Application/Source/System
Pega Applciation-----Response----->External Application/Source/System

5.1 Service-SOAP

In a Service-SOAP, External Application is going to give the request. Based upon request, Pega Application will send the response to External Application/Source/System.

In a Service-SOAP, PEGA Application will provide the WSDL URL or File.

Pega Applciation<-----Request-----External Application/Source/System
Claim ID
Pega Applciation-----Response----->External Application/Source/System
Claim Holder Name,Claim AMount,Claim Reason,Vendor Name

To Implement Service SOAP rule, we need to create below rules.

1. Create Request Parameters

Ex: Claim ID

2. Create Response Parameters

Ex: Claim Holder Name, Claim AMount, Claim Reason, Vendor Name

3. Create Service Package rule

By Using Service Package we can process the request into pega application with the help of Service method like Service SOAP

In a Service Package rule we need to provide the Process Mode(StateLess or StateFull), Access Group and Authentication

Note: If we want to clear the response data from clipboard after completion of the request then we will select the Process Mode to "StateLess".

Note: If we want to Maintain the response data in a clipboard after completion of the request then we will select the Process Mode to "StateFull".

Q) What is use of Access Group in a Service Package?

Ans) Access Group is response to call the rules in a pega Applciation based External Application Request.

Access Group → Application → List of Rulesets → List of Rules.

Q) What is use of Authentication in a Service Package?

Ans) To Secure our Service Request or Security purpose we will maintain the Authentication.

Q) Is it possible to use the same service package rule to different service methods?

Ans) Yes we can use the Same Service Package to multiple Service methods like Service SOAPS

4. Create Parse XML Rule

By using Parse XML Rule we can convert the XML request to clipboard format

5. Create XML Stream Rule

By Using XML Stream rule we can convert the Clipboard response format to xml format.

6. Create Service SOAP rule

Service Tab

In a Service Tab we will map the Request & Response Parameters class name ,Page name and will call the activity to send the response to external application

Request Tab

In a Request Tab we need to call the Parse XML rule to convert the XML request to clipboard format.

Response Tab

In a Response Tab we need to call XML Stream rule to Clipboard response format to xml format.

Fault Tab

In a Fault Tab we will handle the Exceptions

7. Create Activity rule

In a Activity we will write a business logic to send the response to External application.

HDFCB-Int-Clam

RequestPage

ResponsePage

HDFCB-Int-ClaimRequest

Claim ID

HDFCB-Int-ClaimResponse

Claim Holder Name, Claim AMount, Claim Reason, Vendor Name

<http://localhost:8080/prweb/PRSOAPServlet/SOAP/ClaimPKG/Services?WSDL>

Assignments

1.We need to insert the Claim Request Details into PEGA Application

Claim ID Claim Holder Name Claim Reason Claim Amount

2.We need to the Claim Request Details from PEGA Application based Request

Claim ID:244

3.Create Employee Table

Employee ID Employee Name Employee Age Location / Hyderabad

We need to send the list of Email details when we are getting the Location Name Request

5.2 Connect - SOAP

- a. In a Connect-SOAP, PEGA Application is going to give the request. Based upon request, External Application or Source will send the response to pega Application.
- b. In a Connect-SOAP, External Application or Source will provide the WSDL URL or File.

WSDL URL: <http://www.dneonline.com/calculator.asmx?wsdl>

<http://www.thomas-bayer.com/axis2/services/BLZService?wsdl>

Pega Application-----Request----->External Application/Source/System

Complaint ID

Pega Application<-----Response-----External Application/Source/System

Complaint Name, Complaint Date, Complaint Reason, Complaint Holder Name

To Implement Connect SOAP we need to create below rules.

1. Create Request Parameters
2. Create Response Parameters
3. Create XML Stream rule
 - ➔ By using XML Stream rule we can convert the Clipboard request to XML Format
4. Create Parse XML rule
 - ➔ By Using Parse XML rule we can convert the XML response to CLipboard format
5. Create Connect SOAP rule

Service Tab: In a Service Tab we need to provide the EndPoint URL(WSDL URL) and response Time out

Request Tab: In a Request Tab we will call the XML Stream to convert Clipboard request to XML format

Response Tab: In a Resposne Tab we will call the Parse XML to convert the XML resposne to Clipboard Format

Fault Tab: In a Fault Tab we will handle exceptions by using "pySOAPFaultCode, pySOAPFaultReason, pySOAPFaultDetails, pyStatusValue, pyHttpResonseCode" Properties.

6. Create Activity or Data Page rules

Activity:

In a Activity we will call the Connect SOAP rule by using Connect SOAP method. Before Connect SOAP method we wil map the request parameters by using Property-set method or Data Transform.After Connect-SOAP method we will resposne parameters by using Property-set method or Data Transform.

Data Pages:

In a Data Page, as a Source we will Select Connector and call SOAP rule, Request Data Transform, Resposne Data Transform. In a Request Data Transform we will map the Request parameters. In a Resposne data transform we will map response data transform.

5.3 How to handle the Exceptions in Integrations

1. We will validate whether we are sending Correct Request format or not by using Validations or when conditions.
2. We will set log messages success or failure scenarios.
3. While Calling the Connect SOAP or Connect REST rule in a activity then we will handle the exceptions by using "Jump" Conditions.
4. By using "pyHttpResponseCode, pyStatusValue or pySOAPFaultCode, pySOAPFaultReason" properties we can show different labels messages in a UI like success or failure scenarios.

Request XML

```
<?xml version='1.0' encoding='utf-8'?><soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"><soapenv:Body><ns1:getBank xmlns:ns1="http://thomas-bayer.com/blz/">
<ns1:blz>41651770</ns1:blz>
</ns1:getBank></soapenv:Body></soapenv:Envelope>
```

Response XML

```
<?xml version='1.0' encoding='utf-8'?><soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"><soapenv:Body><ns1:
getBankResponse xmlns:ns1="http://thomas-
bayer.com/blz/"><ns1:details><ns1:bezeichnung>Sparkasse
Hochsauerland</ns1:bezeichnung><ns1:bic>WELADED1HSL</ns1:bic><ns1:ort>Br
ilon</ns1:ort><ns1:plz>59929</ns1:plz></ns1:details></ns1:getBankResponse></soa
penv:Body></soapenv:Envelope>
```

5.4 Service REST

XML Format

```
<SampleResponse>
  <FirstName>Kiran</FirstName>
  <LastName>Kumar</LastName>
  <Age>28</Age>
  <Address>Hyderabad</Address>
</SampleResponse>
```

JSON Format

```
{
  "FirstName": "Kiran"
  "LastName": "Kumar"
  "Age": 27
  "Address": "Hyderabad"
}
```

SOAP

- SOAP will support only XML format Request & Responses.
- SOAP will use the more bandwidth to processing Request or Response from Pega Application
- SOAP more secure than REST
- XML format will support Images or Links as a request or response parameters

REST

- REST will support both JSON and XML format Request & Responses.
- REST will use the less bandwidth to processing Request or Response from Pega Application

- REST less secure than SOAP
- JSON format will not support Images or Links as a request or response parameters

Service REST

- In Service REST, External Application is going to give the request in the form of JSON format, Pega application will send the response to external system or application based Upon Request.
- PEGA Application will provide the WSDL URL or File.

PEGA Application<----Request-----External Application
Complaint ID

PEGA Application----Response----->External Application
Complaint Name, Complaint Date, Complaint Reason, Status

To Implement Service REST rule we need to create below rules:

- Create Request parameters
- Create Response Parameters
- Create Service Package rule
In a Service Package rule we need to provide the Process mode, Access group and Authentication
- Create Service REST rule
In a Service REST we will map the request & Response Parameters and Call the Activity to send the response to External Application

Service Tab:

we need to provide the Request & Response Parameters Related class name and Page Name

Methods Tab

Every method we need to map the request and response parameters and call the activity to Send the Response to External System.

a. GET

➔ If we want to send the Response from pega Application based upon External System Request then we can use the "GET" method.

b. POST

➔ If we want to insert the New Record into the pega Application based upon External System Request then we can use the "Post" method.

c. PUT

➔ If we want to update the Existing Record into the pega Application based upon External System Request then we can use the "PUT" method.

d. DELETE

➔ If we want to delete the record from pega application based upon External System Request then we can use the "Delete" method.

e. PATCH

➔ If we want to update the minimal changes into the Existing Record into the pega Application based upon External System Request then we can use the "Patch" method.

5. Create Activity

In a activity we will write a logic to send the response to external application.

Ex: <http://localhost:8080/prweb/api/ComplaintPKG/v1/Complaint/{ComplaintID}>

5.5 Connect REST

- a. In a Connect REST, Pega application is going to give the request, External Application will send the response back to pega application.
- b. External Application will give the WSDL URL.

PEGA Application-----Request----->External Application

Employee ID

PEGA Application<-----Response-----External Application

Employee Name, Employee Age, Employee Salary, Location

To Implement Connect REST rule we need to create below rules

1. Create Request Parameters
2. Create Response Parameters
3. Create Connect REST rule

Service Tab:

In a Service tab We will provide the WSDL url under Resource Path

Method Tab

In Method tab we will map request and response parameters

- ➔ GET: If we want to fetch the data from External System or Application based pega Application request then we can use the GET method.
- ➔ POST: If we want to insert the New record into the External system based pega Application request then we can use the POST method.
- ➔ PUT: If we want to update the existing record into the External system based pega Application request then we can use the PUT method.

- ➔ DELETE: If we want Delete record from External system based pega Application request then we can use the DELETE method.
- ➔ Patch: If we want to Update minor changes into the existing record based pega Application request then we can use the Patch method.

4. Create Data Page or Activity

In a Data Page we will call the Connect REST by using Connector as source in a Data Page and will call request & response data transforms to map request parameters & response Parameters

In a Activity we will call the Connect REST rule by using Connect-REST method in a activity. Before Connect REST method we will map the request parameters by using request data transform or Property -set method.After Connect REST method we will map response parameters by using Response Data Transform or Property-Set method.

Ex: <https://dummy.restapiexample.com/api/v1/employees>

<https://dummy.restapiexample.com/api/v1/employee/1>

<https://gorest.co.in/public/v2/users>

Response:

- 1.Status
- 2.Data (PageList)
- 3.Messages

ASSIGNMENTS/REQUIREMENTS: Display the List of Employee IDs under Dropdown. Based upon selected employee id from above dropdown we need to display the list employee details in a UI

5.6 Dynamic System Settings(DSS)

1. If we want to set the Constant Value (Different in Env to Env (DEV, SIT, UAT, PREPROD, PROD)) then we can create Dynamic System Settings.
2. DSS rule does contains any ruleset version
3. By using @getDataSystemSetting() PEGA OOTB Function we can get the DSS Value

How to Refer DSS value in Connector Rule:

1. Create DSS rule
2. Create Data Transform and get DSS value by using @getDataSystemSetting()

Ex: @getDataSystemSetting(HDFCB, BLZURL)

3. Create Data Page and call the Data Transform
4. Refer the Data Page in Connector Rule

Ex:=DataPageName.PropertyName = D_GetBLZURL.BlzUrl

Application Settings(AS)/(System Settings):

1. If we want to set the Different Constant value to Env to Env then we will Application Settings.
2. Application Settings contains the Ruleset version
3. By using @getRuleSystemSetting() Pega OOTB function we can get AS value
Ex:@getRuleSystemSetting(HDFCB,BLZURL)

Ex: Based upon "pzProductionLevel" property value it will set the Application Setting value.

Dynamic Class Reference(DCR):

If we want to refer class name dynamically in activity method levels dynamically then we can use Dynamic Class Reference(DCR)

1. Create Value Group Property
2. Create Data Transform and Set the Values to Value Group Property
3. Create Data Page and Call the Data Transform
4. Refer the Data Page in Activity method level. Ex:
`D_GetRefClassName.RefClassName(Customer)`

Global Resource Settings(GRS)

Will refer common things (DBs, Email IDs, WSDL URLs,) by creating common referencing table

1. Create Activity to fetch GRS Data – store it in page/pagelist
2. Apply Data Transform and fetch required DBT-name or Email-ID.
3. Create Data Page and Call the Data Transform
4. We can refer D_page to any Activity

6. DATA PAGES (DECLARATIVE PAGES)

By Using Data pages, we can fetch the multiple records from data base table or we can fetch the single record from data base table or we can insert or update the data into data base table or we can connect with external applications by using connectors.

Q): Data Pages will improve the Performance compared to report definition and Activities.

A): Whenever requestor requesting same type of data multiple times, first request it will go the data base and will display the or store the data in a Clipboard. Next Request instead of going to the Data base it will fetch the data from Clipboard under Data pages means it will reduce the network round trips between the Pega application and data base.

By using Data Pages we **can not delete** data from data base table.

Report Definition: Whenever requestor requesting same type of data multiple times, every request it will go the data base.

Activity (Obj-Browse): Whenever requestor requesting same type of data multiple times, every request it will go the data base.

Q) What are the sources available for Page Data Pages?

Ans): Connector, Data Transform, Activity, LookUp

Q) What are the sources available for List Data Pages?

Ans): Connector, Data Transform, Activity, Report Defination

→ From 8.7 Onwards below sources are available for Page and List Data Pages

- 1.Connector
- 2.Data Transform
- 3.Activity
- 4.LookUp
- 5.Report Defination
- 6.Aggrigative Sources

In a Data Page we can call multiple Sources and will execute sources based upon when conditions

If we select the "Aggregative Source" as a source in a Data Page we will call the multiple sources like data transforms, Report Defiantion, Activity and Connectors.

Q) What is the use of "Simulate data source" check box in a Data Page?

when we are connecting external application as a source REST or SOAP by using Data pages, If External System is not working due to internal issues then we can set the Static data by using Data transforms, Activities and report defination.

Q) How to handle the Exceptions in a Data Page?

In a Data Page we have "Post load processing" option is available, there we can call the actcivity to handle the exceptions.

Types of Mode:

1. Read Only: If we don't want to give the option to edit the data pages related data in clipboard then we will select the mode to "Read only".

2. Editable: If we want to give the option to edit the data pages related data in clipboard then we will select the mode to "Editable".
3. Savable: if we want to insert or update the data into data base table then we will select the mode to "Savable".

Types of Scopes

1. Thread:

- Thread Level Data pages is a nothing but Work Object level. Whatever data is available under Thread Level Data page, it will accessible to only that particular Work Object(Thread).
- After Creating a Work Object, it will load the Thread level Data Pages.

Order ID Get Order Info

Order Name Order Price

1---2-Order Info---3--Customer Info---4

Ex: Agent --->Thread1 (Work object)--->D_GetOrderInfo OrderID: 1213

Agent --->Thread2 (Work object)--->D_GetOrderInfo OrderID: 4545

2. Requestor:

- Requestor Level Data Page is a nothing but Operator level Data Page. Whatever Data is available under Requestor level Data page we can access to Multiple Threads (Work Objects).
- Requestor Level Data page will load at the time of login into the application.

Note: For Requestor or Thread Level Data pages, it will consider the Requestor Access group at the run time

Ex: Agent or CSR or BackOffice User → Thread1,Thread2,Thread3

D_AgentInfo or D_CSRInfo

3. Node:

- Node Level Data Page is a nothing but Application-level Data page. Whatever data is available under Node data pages, every requestor can able to access.
- Node Level Data Page will load at the time of Start the Server
- Node Level Data pages related data we cannot edit because node level data pages related data sharable to all requestor whoever login into the that particular node.it will give Data mismatches if we give the Editable option to Node level Data Pages.

Note: Editable Data Pages related data we can see in a clipboard under "User Pages".

Q) What Type of Data pages we need to provide the Access group?

For Node Level Data Pages we need to provide the Access group. Access group will responsible to call the rules in pega whatever source (Data transform or activity or connector or Report Defination) called in a Node Level Data Page.

Ex: Customer1----->D_PolicyList

Customer2----->D_PolicyList

Customer3----->D_PolicyList

Q) How to flush or clear the Data pages in pega

1. In a Data page under Load Management tab we have "Clear Data Page" option is available to flush or remove the Data Page data from Clipboard.
2. By using "pzFlushDataPage" pega ootb activity we can flush or remove the Data Page data from Clipboard.
3. By using `@(Pega-RULES:DeclarePages):pzDeleteAllInstancesDeclarePages("Data Page Name")` pega ootb function we can flush or remove the Data Page data from Clipboard.

Q): what is the use of "Refresh strategy" in a data page?

If we want to reload the Data Pages by using some time intervals like days or hours or minutes or seconds or based upon when conditions then we can specify the days or hours or minutes or seconds or when conditions in a data page under load management tab.

Types of Refresh strategy:

1. Reload once Per Interaction: It will reload the data page at the first interaction.
2. Do not reload when: We can call when condition to reload the data page
3. Reload if older than: We can specify the days or hours or minutes or seconds to reload the data-page

at 9am 9.30am 9.45am

1hr 10am

6.1 Types of Data Pages:

1. **Page Data Page:** By Using Page Data Page we can fetch the single record from data base table by passing primary key value.

D_GetCustomerInfo[CustomerID:Param.ID]

D_GetCustomerInfo[CustomerID:"111"]

D_GetCustomerInfo[CustomerID:.CustomerID]

D_GetCustomerInfo[CustomerID:.CustomerPage.CustomerID]

D_GetCustomerInfo[CustomerID:111].CustomerName

D_GetCustomerInfo[CustomerID:111].CustomerAge

D_GetCustomerInfo[CustomerID:111].Location

2. **List Data Page:** By Using List Data Page we can fetch the multiple records from data base table.

D_GetCustomerRecords[ID:.CustomerID,Name:.CustomerName,Age:.Age,Loc:.Location].pxResults

D_GetCustomerRecords[Age:"20",Loc:"Hyderabad"].pxResults

3. **Savable Data Page:** By Using Savable Data Page we can fetch insert or update the data into data base table.

To configure Savable Data-Page:

- i. Create Data Transform and set the values
- ii. Create Data Page and Select mode to "Savable", Source to Data Transform

- iii. Call the Savable Data Page in a Flow Action under Post Processing or In a Flow by using Save Data Page automation shape.

4. Key Accessed Data Page:

- Key Accessed Data Page will fetch the data based upon Keys whatever we are passing from Property Level.
- Key Accessed Data page will not hit the data base every time to fetch the data
- Key Accessed Data page will not maintain the instances in a clipboard
- Key Accessed Data page will refer in Property level like page or pagelist properties. we cannot refer in a Section as source or other rules.

Note: Only List Data page we can pass the Keys.

- Step1: Create List Data Page and Select the Keys
- Step2: Create Page or PageList property and Refer Key Accessed Data Page
- Step3: Refer the Page or Pagelist properties in a Section

Ex: Policy Type-----Insurance Company----->Policies

5. Parameterized Data Page:

- Parameterized Data Page will fetch based upon parameters from UI.
- Parameterized Data page will hit the data base every time to fetch the data whatever parameters we are passing.
- Parameterized Data page will maintain the instances in clipboard
- we can refer Parameterized Data page in a section and other rules also.

Ex: Policy Type-----Insurance Company----->Policies

xxx

yyy

zzz

6.2 Types of Data Access Patterns in PEGA:

1. Alias Pattern

2. System of Record Pattern

3. SnapShot Pattern

4. Reference Pattern

5. Keyed Access Pattern

7. REPORT DEFINITION

If we want to fetch the multiple records from Data base table or Based upon Filter conditions if we want to fetch the Results from data base table or if we want to join the two or more than two tables or if we want to display the data in the form of "charts or Table format" or If we want to schedule the reports based upon some time intervals then we can go for Report Defination.

→ By using "**pxRetrieveReportData**" Pega OOTB activity we can call the report definition in a activity. we need to pass the "pyReportName, pyReportClass and pyPageName" parameters.

→ By using "**pxCallRetrieveReportData**" Pega OOTB activity we can call the report defination in a data transform. we need to pass the "pyReportName, pyReportClass and pyPageName" parameters.

Right click on class → Create → Reports → Report Defination

7.1 Types of Reports

1. List View:

If we want to display the data in the form of Table format then we can go for List View

2. Summary View:

If we want to display the data in the form of "Charts or Graphical Representation" then we can go for Summary View

3. Report Definition:

If we want to display the data in the form of Table format or "Charts or Graphical Representation" then we can go for Report Defination.

Report definition: Student RPDef [Available]
 CL: HDFC-Data-StudentInfo ID: StudentRPDef RS: Loan:01-01-01

This record has 1 severe or moderate warning (including 1 unjustified) and 1 info warning [View](#)

Query Chart Report Viewer Data Access Parameters Pages & Classes Test cases Specifications History

Edit columns

Column source	Column name	Summarize	Sort type	Sort order
.StudentID	Student ID	<blank>	<blank>	
.StudentName	Student Name	<blank>	<blank>	
.Age	Age	<blank>	<blank>	
.DepartmentID	Department ID	<blank>	<blank>	

[Add column](#)

Edit filters

Filter conditions to apply

1

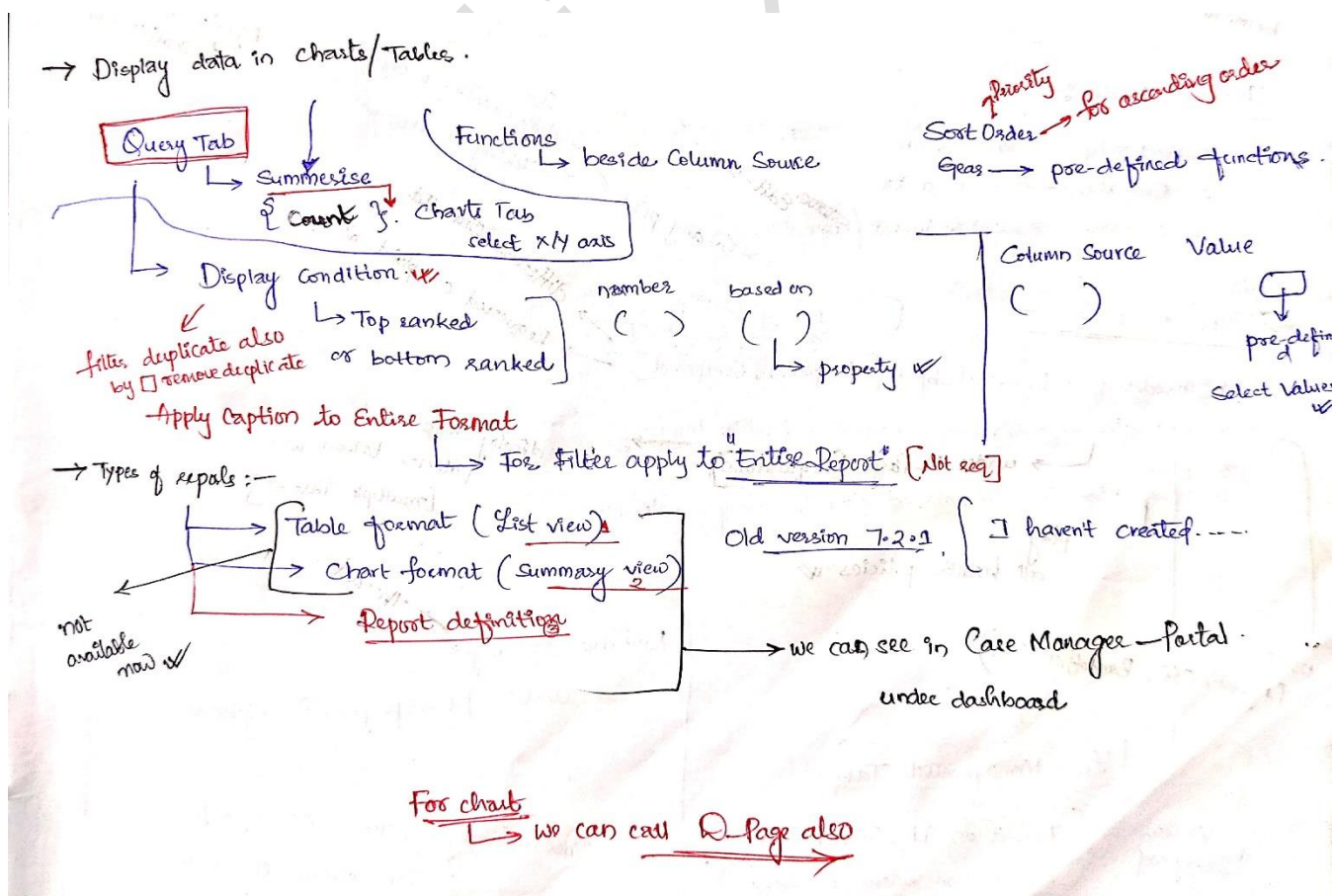
Condition	Caption	Column source	Relationship	Value
1	Id	.DepartmentID	Is equal	Param.ID

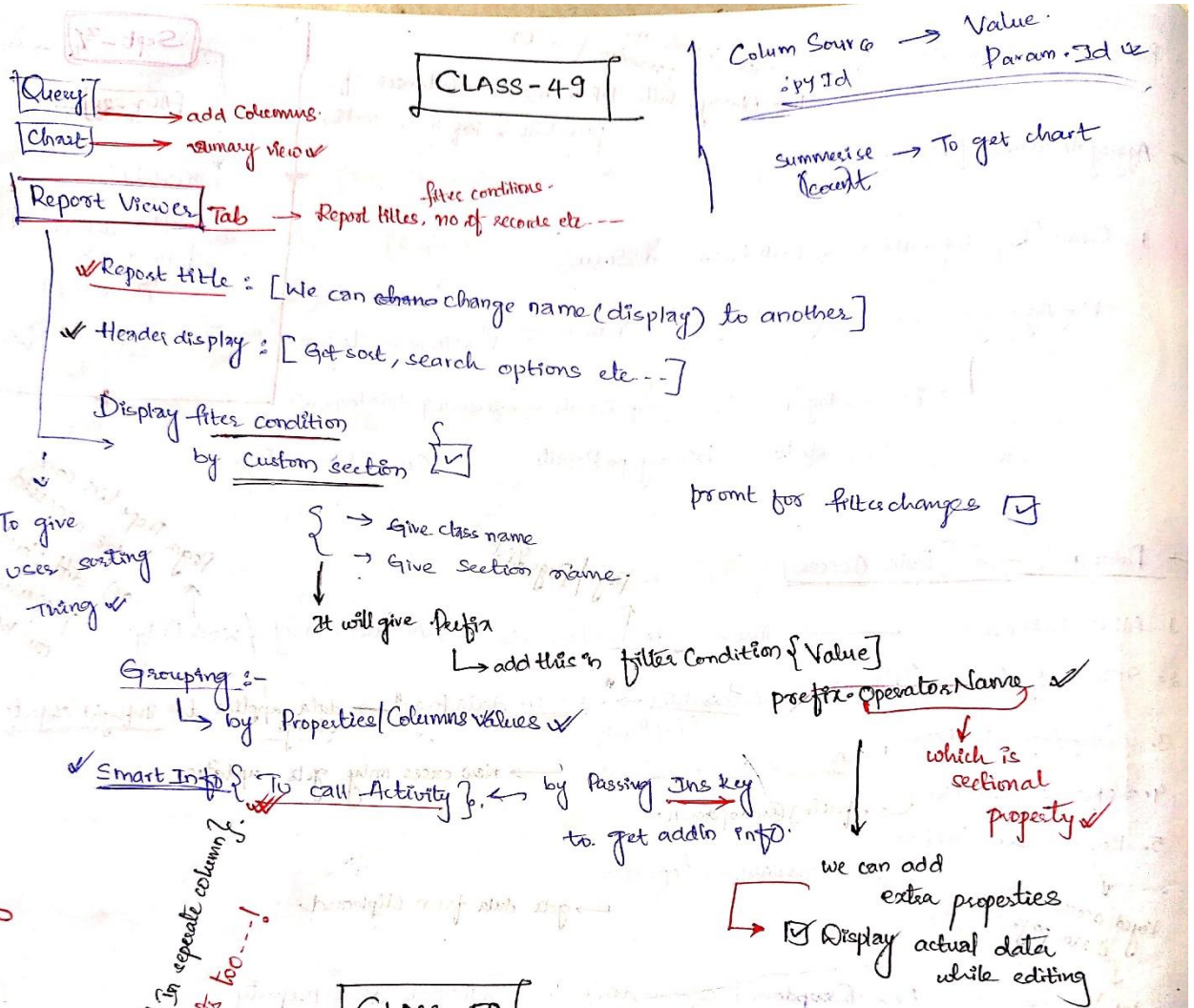
[Add filter](#)

Filter caption settings

Apply caption to entire condition

☐ Remove duplicate rows





Case ID	Operator Name	Balance Amount	Update Date Time
C-1212	Kiran	200	10.10am
C-1212	Kiran	200	10.12am
C-1212	Kiran	200	10.13am

Report Definition

Ex: Screen

SearchCreditCardCase --Link

Search Case Details

Start Date: End Date Operator Name: Status:

7.2 How to Expose or Optimize the Properties in PEGA

Exposing or Optimizing a properties is nothing but "Creating New Column" in the Data base table.

Note: If we are not Optimizing the Properties then pega it will fetch the data from "pzPVStream(BLOB) Column".

```
SELECT  "PARENTPC0".L1    AS  "pyID"    ,  "PARENTPC0".L2    AS
"pxCreateDateTime" , "PARENTPC0".L3 AS "pyStatusWork" , "PARENTPC0".L4
AS  "FeedbackComments" , "PARENTPC0".L5 AS  "OrderPurchaseDate" ,
"PARENTPC0".L6 AS "pzInsKey" , "PARENTPC0".L7 AS "pzRank" FROM (
SELECT  "PC0"."pyid" AS "11" , "PC0"."pxcreatedatetime" AS "12" ,
"PC0"."pystatuswork" AS "13" , data.pr_read_from_stream('FeedbackComments',
"PC0".pzInsKey, "PC0".pzPVStream) AS "14" ,
data.pr_read_from_stream('OrderPurchaseDate', "PC0".pzInsKey,
"PC0".pzPVStream) AS "15" , "PC0"."pzinskey" AS "16" , ROW_NUMBER() OVER
(ORDER BY "PC0"."pxcreatedatetime" DESC NULLS LAST) AS "17" FROM
data.pc_HDFCB_India_Sale_Bank3_Work "PC0" WHERE "PC0"."pxobjclass" = ? )
"PARENTPC0" WHERE "PARENTPC0".L7 <= ? ORDER BY 7 ASC
```

1st way:

From Dev Studio → Right click on that particular Property → Select Optimize for Reporting → Click on Next → Click on Next → Finish

2nd way:

We need to run the "Alter Table" DB Script in a Data base level

Syntax: ALTER TABLE table_name ADD column_name datatype;
orderpurchasedate

Ex: ALTER TABLE data.pc_HDFCB_India_Sale_Bank3_Work ADD
orderpurchasedate TIMESTAMP(29)

Ex: Customer Table

Customer ID Name Age Location Village State pzPVStream(BLOB) Column

1111 Raju 28 Chennai MainRoad,Kukatpally,TS,504232

Village State

Declare Index:

By Default, Pagelist property related data will be stored into "pzPVStream" blob colum.
To fetch the Page list Property related data from data base table then we will create
seperate "Index" table and declare index rule in pega.

1. Create Index table with Properties

Note: primary keys for Index table: pxInsIndexedKey, pxIndexCount,
pxIndexPurpose

Ex: Index_Dependent Table Name Age location Relation

2. Create Index Class in pega and Properties

Ex: Index-Dependent class Name Age location Relation

3. Create Data base Mapping rule and provide Class name, Data base name and table name

4. Create Declare Index rule and Map the PageList Source properties and Index Table Properties.

5. Create Report Definition and Refer the Declare Index under "Data Access" tab to get pagelist data from data base table.

7.3 How to Join the two or More than two data base tables in pega

By using Report Definition, we can join the two or more than two data base tables in Pega

In a Report Definition, we have different options like "Class Joins, Declarative Joins, Associations and sub reports" to join the two data base tables in Pega.

In a Report Definition under "Data Access" tab we have "class Join" option is available. There we need to specify the "Prefix name, Class Name, Type of Join and Joined Conditions".

Work List related Assignments data will be stored into "pc_Assign_WorkList" Table.

Work Queue or Work basket related Assignments data will be stored into "pc_Assign_WorkBaket" Table.

"pc_Assign_WorkList" Table belongs "Assign-WorkList" Class

"pc_Assign_WorkBasket" Table belongs "Assign-WorkBasket" Class

Types of Class Joins:

1. Include a Matching Rows(Inner Join): it will fetch the common Records between Data base tables

2. Include All the rows in this class (Left Outer Join): it will fetch all the records from left data base table and common records between Data base tables

3. Include All the rows in joined Class (Right Outer Join): it will fetch all the records from Right data base table and common records between Data base tables

Q) How to Join the "Work" Table and Assignment Tables?

By using "pxRefObjectKey" property of Assignment table and "pzInsKey" property of Work Table we can join the "Work" Table and Assignment Tables.

Associations:

Associations is a nothing but rule. By using "Associations" we can join the two data base tables in pega.

By using Associations rule we can achieve the reusability. Whatever we created the Associations rule we can reuse into the different report definitions of that particular class or Different class.

Note: "pxWorklistAssignments, pxWorkbasketAssignments, pxCreateOperator" pega ootb Associations rule we used in our projects.

Goto Record Explorer → Create → Data Model → Associations

Sub Reports:

Sub Reports is a nothing but "Query within the Query" means calling a report definition within the report definition.

If we want to filter results in a main report based upon sub report filter condition results then we can use the "Sub Reports".

Ex: X Table Company

Wipro

Y Table Organization

Wipro

Joined Condition Company = Organization

Work table

Operator Table

Work Table

Assignments Table

7.4 How to Schedule the Reports to Business Users in PEGA

1. Category Rule: It will define the Category of the Report.
2. Shortcut rule: It contains the Report Definition class, Report Definition and category name.

Note: Pega will send the Reports from "Default" Email Account.

DrillDown Report:

-->In a Main Report, If we select any one Property as Aggrigative Function like Count, Max, Min..etc. Under Report Viewer we can call our own Custom Dripdwn report while clicking the any row in a Main Report.

AGGREGATIVE PROPERTIES:

1. ValueList:

- a. ValueList means single Property with multiple Values
- b. ValueList will follow the Order while storing the values.
- c. In a ValueList, Index should be "Integer"

Ex: PhoneNumber(1):977334343

PhoneNumber(2):987343535

PhoneNumber(3):845774545

2. ValueGroup

- a. ValueGroup means single Property with multiple Values
- b. ValueGroup will not follow the Order while storing the values.
- c. In a ValueGroup, Index should be "Alphanumeric"

Ex: EmailID(Personal):xxxx@gmail.com

EmailID(Office):yyyy@wipro.com

EmailID(FB):zzz@yahoo.com

3. Page

- a. Page Property means Single Page with multiple property Values
- b. While Creating a Page property we need to specify the Page Definition.

Ex:CustomerPage.CustomerName:Kiran

CustomerPage.CustomerAge:37

CustomerPage.Location:Hyderabad

4. PageList

- a. Page List means List of Pages with Multiple Property values.
- b. While Creating a PageList property we need to specify the Page Definition.
- c. PageList will follow the order to store the values.
- d. In a PageList, Index Should be "Integer"

Ex: .CustomerList(1).CustomerID:35353

.CustomerList(1).CustomerName:Ramesh

.CustomerList(1).CustomerAge:35

.CustomerList(2).CustomerID:353

.CustomerList(2).CustomerName:Raju

.CustomerList(2).CustomerAge:53

5. PageGroup

- a. Page Group means List of Pages with Multiple Property values.
- b. While Creating a PageGroup property we need to specify the Page Definition.
- c. PageGroup will not follow the order to store the values.
- d. In a PageList, Index Should be "Alphanumeric"

Ex: .AddressList(Current).AddressLine1:Main Road

.AddressList(Current).AddressLine2:SR Nagar

.AddressList(Current).State:TS

.AddressList(Current).ZipCode:400024

.AddressList(Office).AddressLine1:Main Road

.AddressList(Office).AddressLine2:Kukatpally

.AddressList(Office).State:TS

.AddressList(Office).ZipCode:400023

Types of Indexes:

- 1.(<APPEND>):it will insert the value into property in a higher index position.
- 2.(<LAST>):it will replace the higher index position value
- 3.(<INSERT>):it will insert the value into the specified index position
- 4.(<PREPEND>):it will insert the value into the first position
- 5.(<CURRENT>):it will specify the Current Loop Position

7.5 How to Connect with External Data base Table in PEGA

St:1 Create a Data base rule and Provide Data base name,JDBC url,User name and Password

Ex:Oracle 11g

Goto Records Explore → SysAdmin → Data Base

St:2 Create a Data base Table by using "Create Database Script" in a Data base level.

Ex: CREATE TABLE JPMC_USA_Sale_Bank1_Work_Claim (

ClaimID int,

LastName varchar(255),

FirstName varchar(255),

Address varchar(255),

City varchar(255)

);

St:3 Create Data base Class and Properties in pega application

Ex: JPMC-USA-Sale-Bank1-Work-Claim

ClaimID, LastName, LastName, Address, City

St:4 Create Data base Mapping rule and provide the Data base name,Schema name and Data base table name.

Ex: JPMC_USA_Sale_Bank1_Work_Claim Data base name,PEGADATA Data base name

St:5 Create Connect SQL rule and Write a DB Scripts or We can create Data pages,Report definition or Obj Methods in a activity to fetch the data from External Data base Table

Open Tab:

If we want to fetch the single record from data base table then we need to write a Query under Open Tab.

Ex: Select * from {Class: JPMC-Data-Customer} where CustomerID=111

Browse Tab:

-->If we want to fetch the Multiple record from data base table then we need to write a Query under Browse Tab.

Ex: Select * from {Class: JPMC-Data-Customer}

or

Select * from data.pr_JPMC_Data)Customer

Save Tab

-If we want to insert the records into data base table then we need to write a Query under Save Tab.

Delete Tab:

If we want to Delete the record from data base table then we need to write a Query under Delete Tab.

St:6 Create Activity by using RDB methods.

RDB-List:

→ By using RDB List method in a activity,we can fetch the Multiple records from External data base table.

RDB-Open:

By using RDB Open method in a activity,we can fetch the Single record from External data base table.

RDB-Save:

By using RDB Save method in a activity,we can insert the records into the External data base table.

RDB-Delete:

By using RDB Delete method in a activity,we can Delete the record from External data base table.

Note:

C:\PRPCPersonalEdition\pgsql\pgAdmin 4\bin →

Ex1:X Work Class-----Data----->AAA Data Base Table

Y Work Class-----Data----->AAA Data Base Table

New Requirement

X Work Class-----Data----->PPP Data Base Table

Y Work Class-----Data----->HHH Data Base Table

OBJ METHODS:

- 1) By using Obj Methods we can connect with Internal Data base tables and External data base tables.
- 2) By using Obj Methods we can fetch the unexposed Properties(Column) related data from Data base table.
- 3) If we use the Obj methods,Pega itself will generate the DB(Data base)Scripts at the run time.
- 4) we can use the Obj-Open-By-Handle and Obj-Delete-By-Handle methods to internal data tables
- 5) When we use the Obj methods, we need Data base mapping rule.
- 6) If we use the Obj Methods we can not call the Obj Methods.

RDB METHODS:

- 7) By using RDB Methods we can connect with Internal Data base tables and External data base tables.
- 8) By using RDB Methods we can not fetch unexposed Properties(Column) related data from Data base table.
- 9) If we use the RDB methods, We need to write the DB(Data base)Scripts at the run time.
- 10) we can not use the Obj-Open-By-Handle and Obj-Delete-By-Handle methods to external data tables
- 11) When we use the RDB methods , Data base mapping is optional because we will create connect sql rule and write the queries.
- 12) If we use the RDB Methods then we can write complex DB Queries and call the Stored Procedures.

8. LOOPING & OBJ- METHODS

```
For(i=0, i<7, i++)
```

```
{
```

```
    Inside the loop
```

```
}
```

PyForEachCount →

1. Property-Set

For Each Embeded Page

```
Primary.CustomerList(<APPEND>).CustomerID
```

```
Primary.CustomerList(<LAST>).CustomerName
```

```
Primary.CustomerList(<LAST>).CustomerAge
```

```
Primary.CustomerList(<LAST>).Location
```

2. Page-Copy

```
Primary.CustomerList(<APPEND>)
```

Note:-By using "pxRetrieveReportData" PEGA OOTB Activity we can call the report defination in a activity. we need to pass the "pyReportName, pyReportClass, pyPageName" parameters

Note:-If we want to refer or call the data page in a activity then we need to specify the Data Page name and class name in a Pages & classes.

1. Obj-Open:

By using Obj-Open method in a activity we can fetch the single record from data base table by passing primary key value

2. Obj-Open-By-Handle:

By using Obj-Open-By-Handle method in a activity we can fetch the single record from data base table by passing "pzInsKey" value

Q) What is the format of "pzInsKey" value?

ClassName PrimaryKey

Ex: HDFCB-INDIA-SALE-BANK3-WORK C-10003

3. Obj-Delete:

By using Obj-Delete method in a activity we can Delete the record from data base table.

4. Obj-Delete-By-Handle:

By using Obj-Delete-By-Handle method in a activity we can Delete the record from data base table by passing "pzInsKey" value

5. Obj-Save

By using Obj-Save method in a activity we can insert or update the data into the data base table

6. Obj-Save-Cancel

By using Obj-Save-Cancel method in a activity we can revert the Obj-Save method changes from data base table if Exceptions occurs while saving

7. Rollback:

By using "RollBack" method in a activity we can revert the Obj-Save method changes from data base table if Exceptions occurs while saving

8. Commit:

By using "Commit" method in a activity we can save (Commit) the data into data base table permanently.

9. Write-Now:

If we check the "Write-Now" check box under Obj-Save method in a activity then we can save(Commit) the data into data base table immediately.

10. Obj-Refresh-And-Lock:

By using "Obj-Refresh-And-Lock" method in a activity we can aquirie the lock on the work object while opening the work object.

Ex: Flow

1---2-Sravan--3-----4

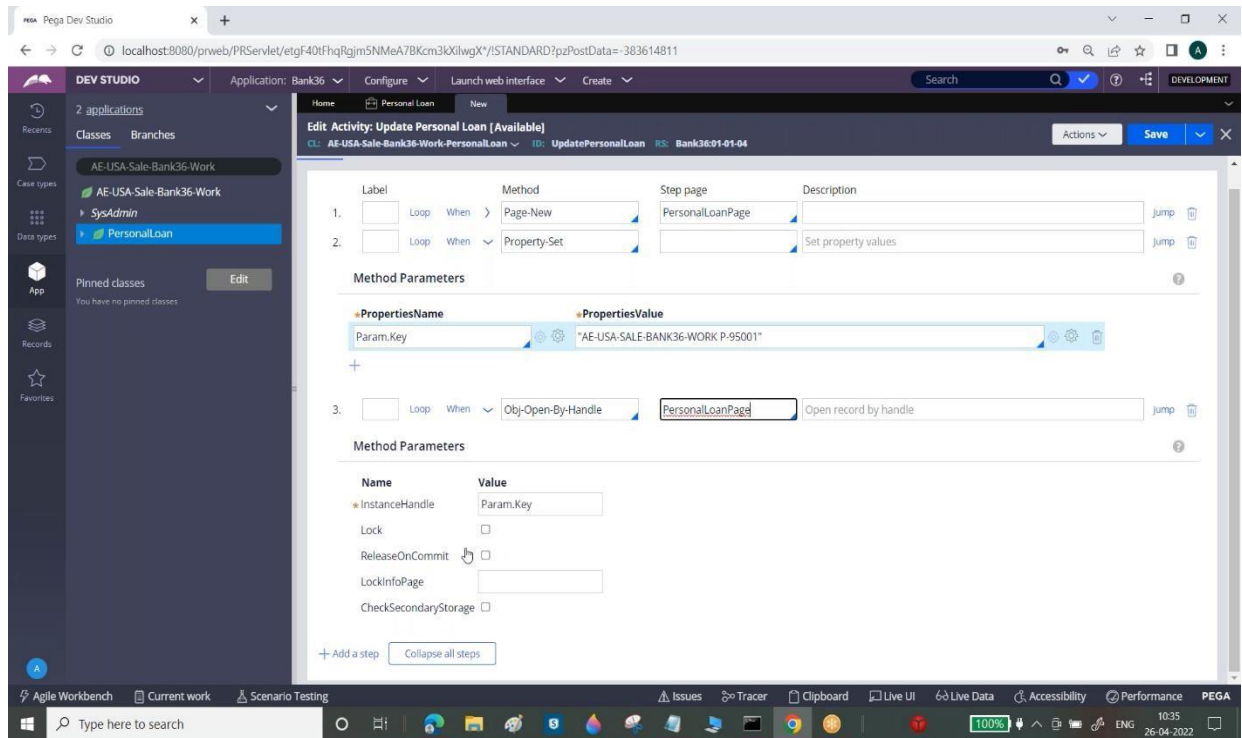
Assign-WorkList

Assign-WorkBasket

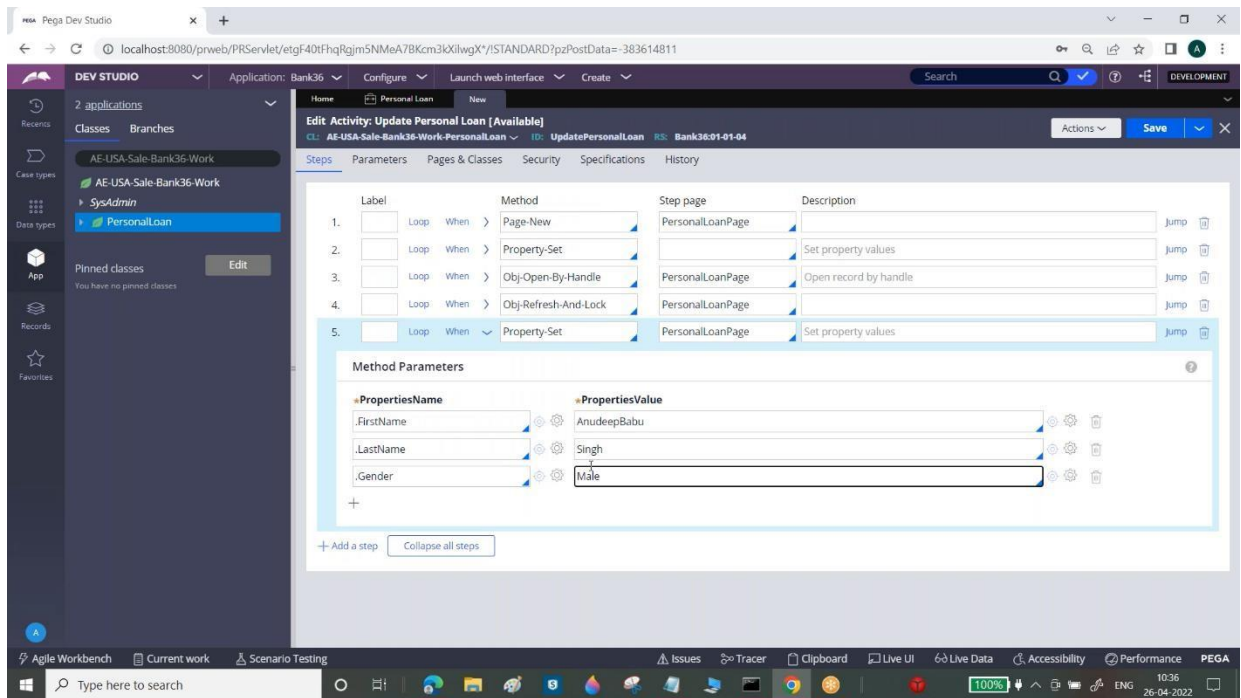
Requirement:

Q) I've particular workobject data I need to update it like some details and statuswork Pending to open, how do you achieve?

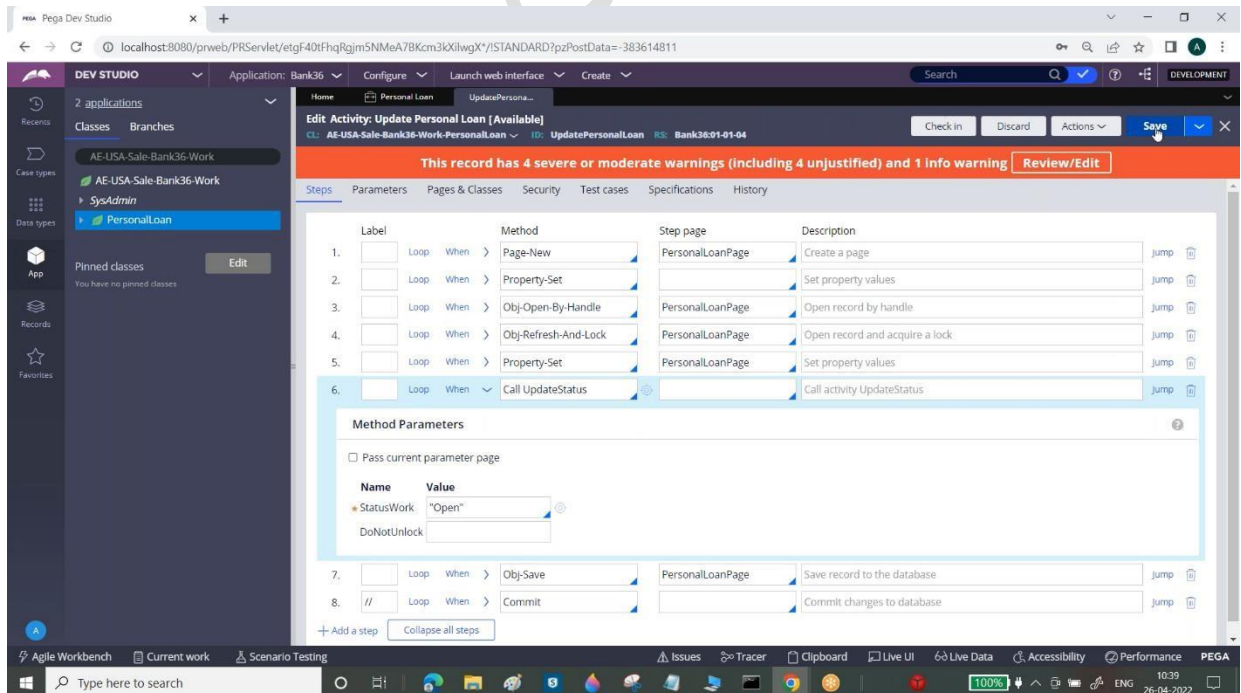
A) I'll create stand alone activity in that work class by using obj-open-by handle and by passing pzInsKey value I'll set and then commit.



We passed the pzInsKey value and will set the values as below



*Call **UpdateStatus**, using this OOTB function it will update the status of work object*



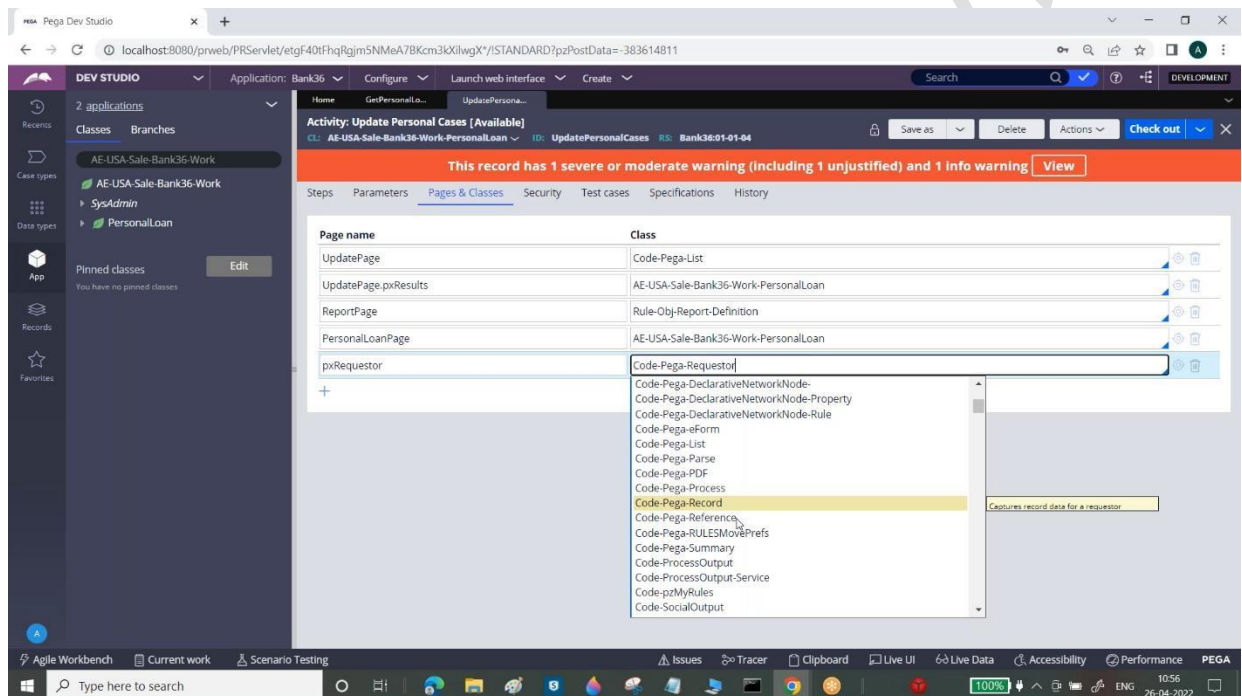
Then save/commit it

Requirement 2:

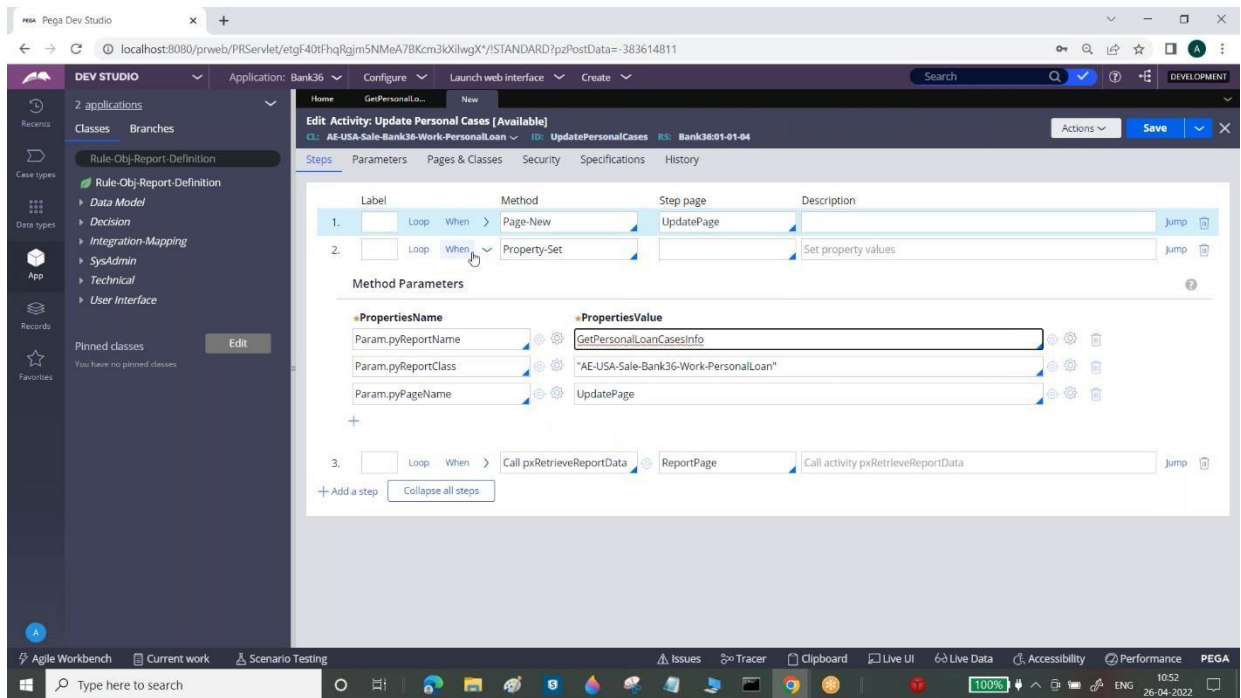
Q) How do you update multiple work objects data like setting data and status work change?

A) I'll create a Report definition in my class and fetch the records based on conditions and filters, I'll create an activity and call the report using "pxRetrieveReportData" and loop them one by one set the values and commit.

Step 1: Mention the pages and classes

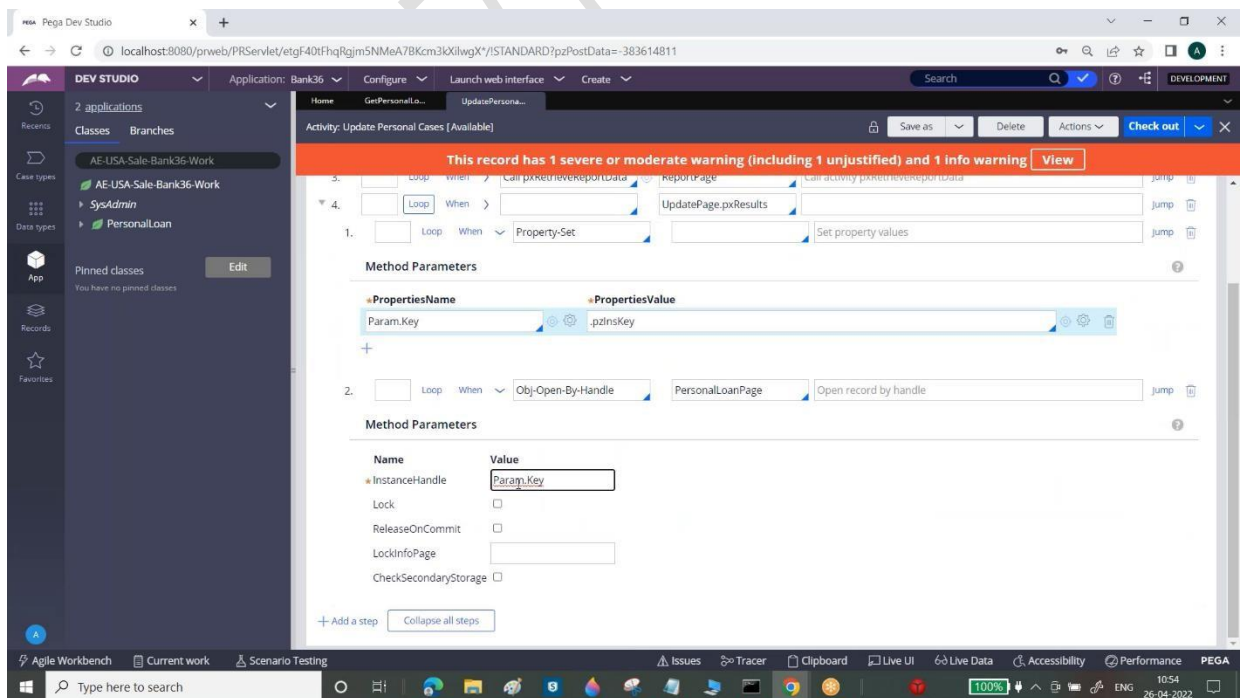


Step 2: Page-new → Propertyset(pyReoprtName, class,page) → call pxRetrieveReportData



After fetching the results loop them and set, the values by opening it with open-handle

Step 3: Set the looping values to param.key(pzInsKey)



Step 4: Set the values you wanted after opening

The screenshot shows the Pega Dev Studio interface for configuring the 'Update Personal Cases' activity. The 'Steps' tab is selected, showing a list of steps with their labels, methods, step pages, and descriptions. A warning banner at the top indicates 1 severe or moderate warning and 1 info warning. The 'Method Parameters' section shows properties like PersonalLoanPage.FirstName, LastName, Age, and Status, with values being set.

Label	Method	Step page	Description
1. Loop When	Page-New	UpdatePage	Create a page
2. Loop When	Property-Set	UpdatePage	Set property values
3. Loop When	Call pxRetrieveReportData	ReportPage	Call activity pxRetrieveReportData
4. Loop When	UpdatePage.pxResults	UpdatePage.pxResults	
1. Loop When	Property-Set		Set property values
2. Loop When	Obj-Open-By-Handle	PersonalLoanPage	Open record by handle
3. Loop When	Property-Set		Set property values

Method Parameters

PropertiesName	PropertiesValue
PersonalLoanPage.FirstName	"Arjun"
PersonalLoanPage.LastName	"Reddy"
PersonalLoanPage.Age	25
PersonalLoanPage.pyStatusWork	"Open"
PersonalLoanPage.pxUpdateDateTim	@CurrentDateTime

Step 5: COMMIT>> Done.

The screenshot shows the Pega Dev Studio interface for configuring the 'Update Personal Cases' activity. The 'Steps' tab is selected, showing a list of steps with their labels, methods, step pages, and descriptions. A warning banner at the top indicates 1 severe or moderate warning and 1 info warning. The 'Method Parameters' section shows properties like PersonalLoanPage.FirstName, LastName, Age, and Status, with values being set.

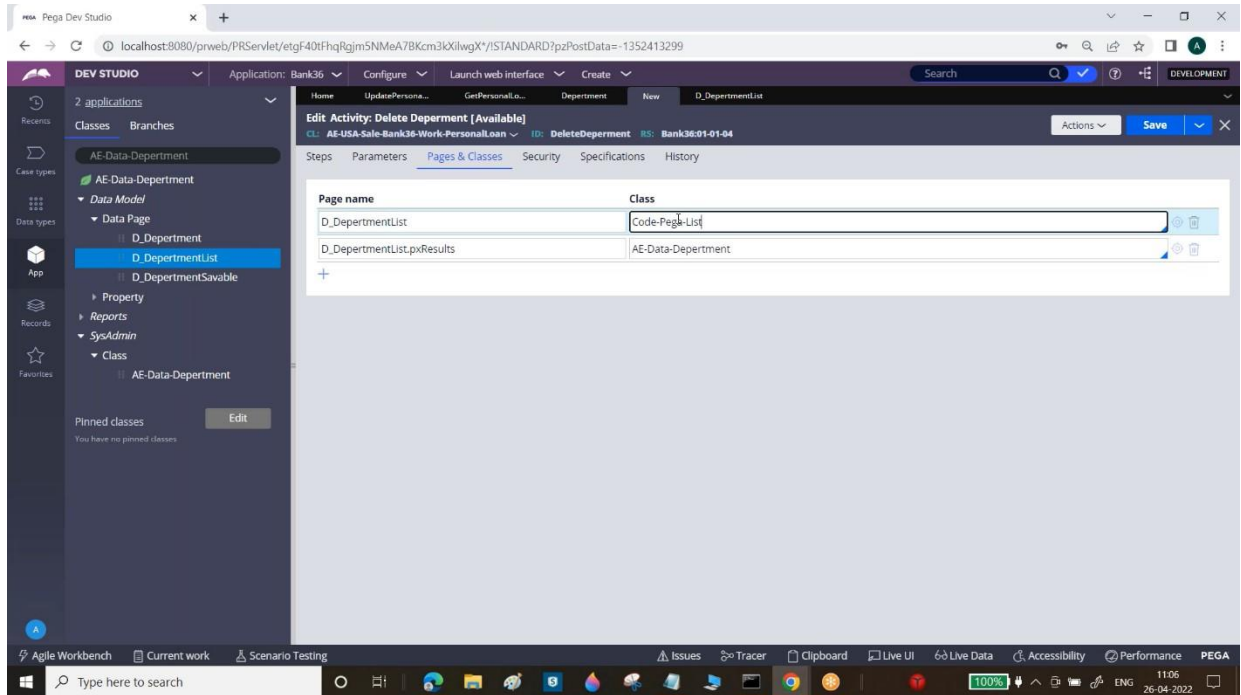
Label	Method	Step page	Description
1. Loop When	Page-New	UpdatePage	Create a page
2. Loop When	Property-Set	UpdatePage	Set property values
3. Loop When	Call pxRetrieveReportData	ReportPage	Call activity pxRetrieveReportData
4. Loop When	UpdatePage.pxResults	UpdatePage.pxResults	
1. Loop When	Property-Set		Set property values
2. Loop When	Obj-Open-By-Handle	PersonalLoanPage	Open record by handle
3. Loop When	Property-Set		Set property values
4. Loop When	Obj-Save	PersonalLoanPage	

Method Parameters

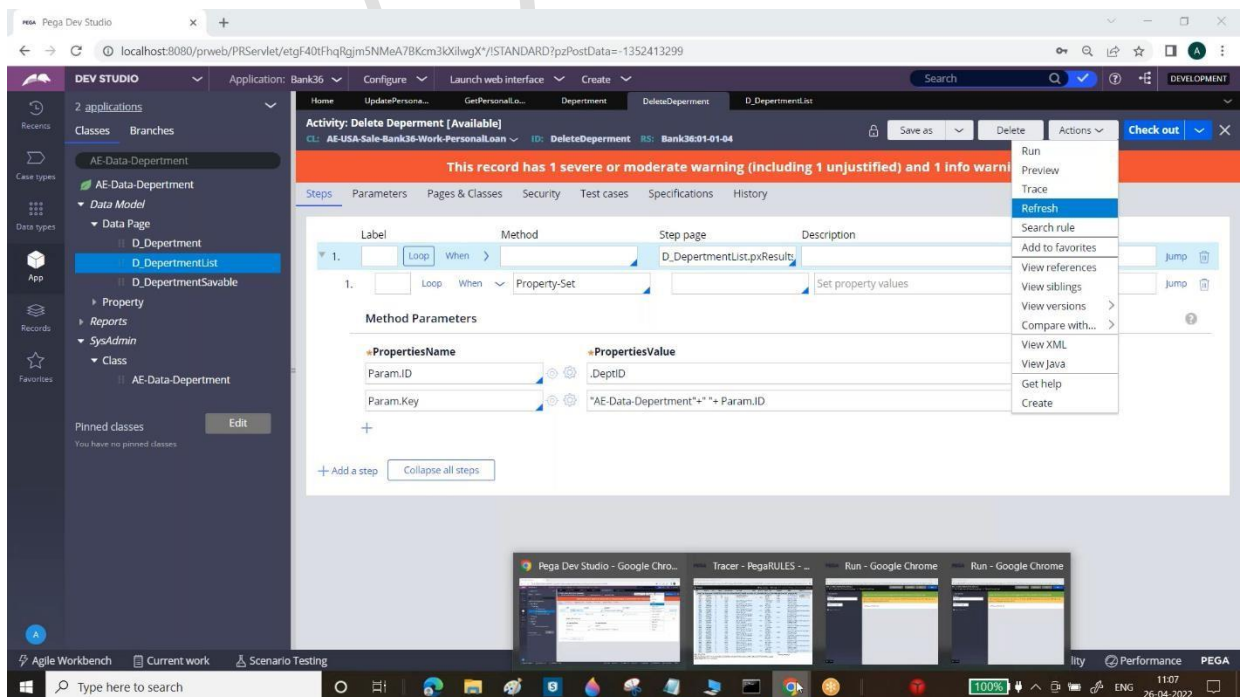
PropertiesName	PropertiesValue
PersonalLoanPage.FirstName	"Arjun"
PersonalLoanPage.LastName	"Reddy"
PersonalLoanPage.Age	25
PersonalLoanPage.pyStatusWork	"Open"
PersonalLoanPage.pxUpdateDateTim	@CurrentDateTime

Requirement:

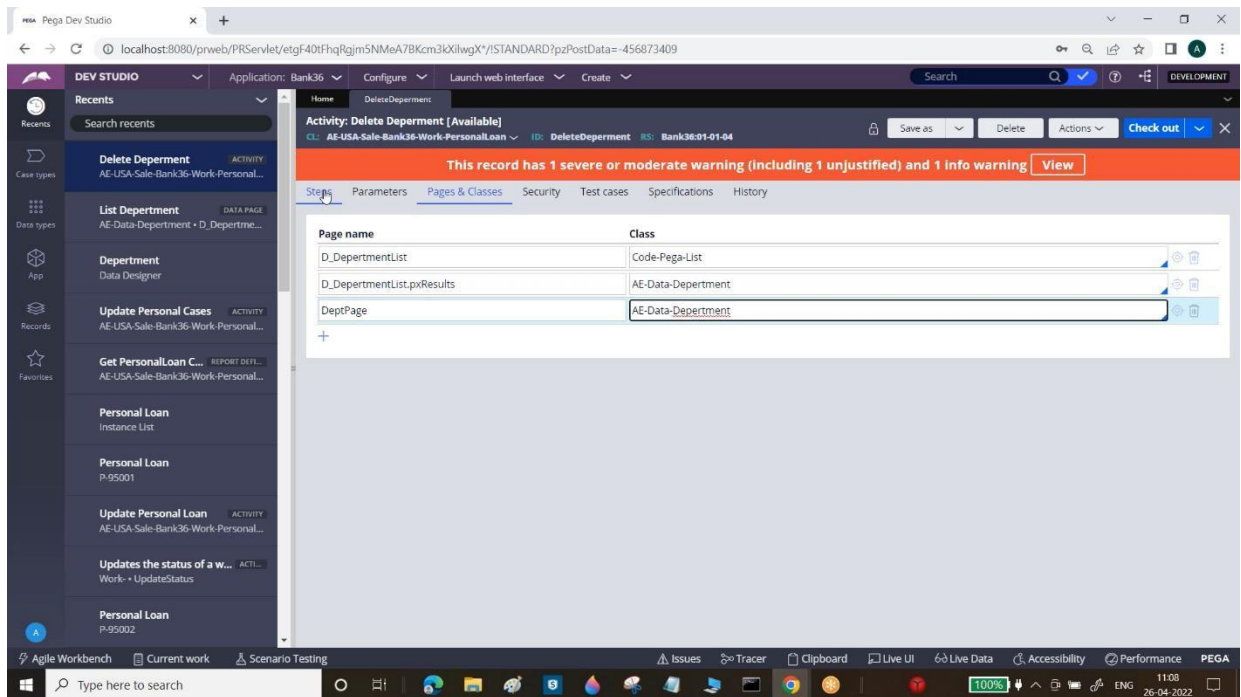
Q) How do you delete all the records from the table? A) By passing pages and classes and looping the results and passing pzInsKey to Obj-Del-handle



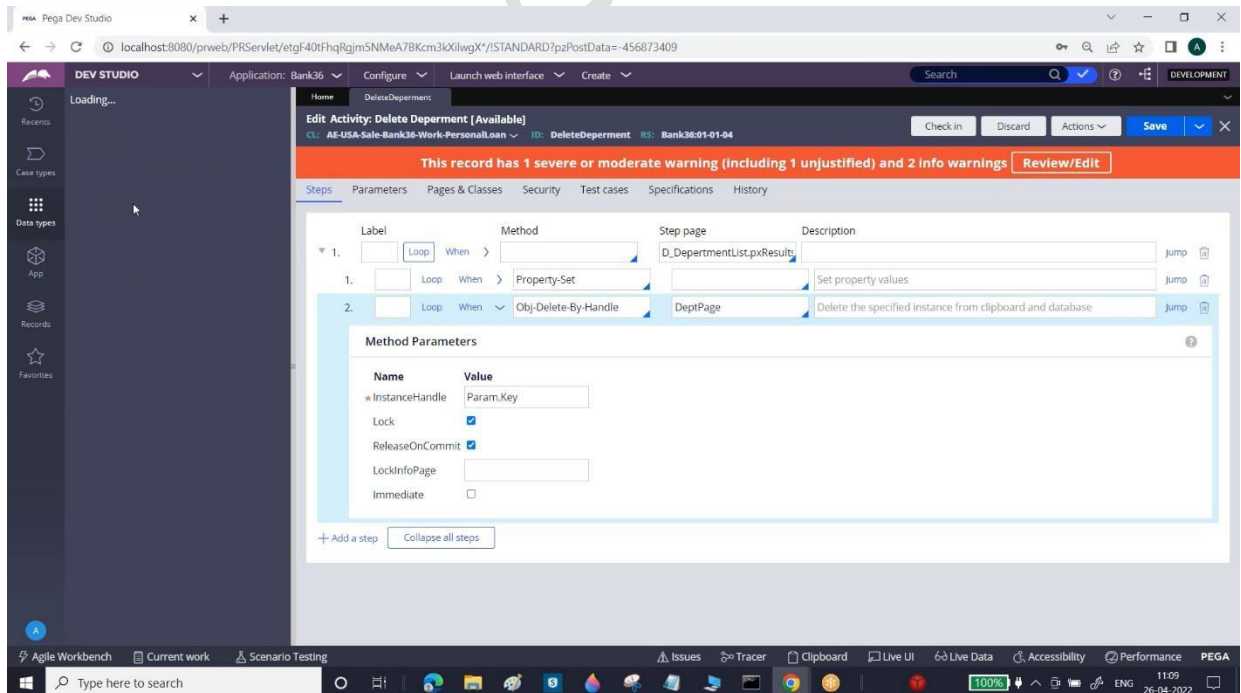
Created a pzInsKey and will pass it to del-handle



Here, you wanted to del the data-department therefore do mention in pages and classes



Then use del- handle and pass the pzInsKey>> run>> Done.



Scenarios:

Q) Table contains list of records fetch those records with conditions and update?

I'll create a report with conditions/filter conditions and call it in activity (or) I'll fetch using obj-browse and fetch the records (or) I'll create data page and as a source I'll call report definition and call the data page in the activity, Inside the loop I'll loop each and every record by obj-open-by-handle by passing pzInsKey Value>> property-set method and update the values/ properties after that property-set, Save and commit the data, DONE.

Q) Update the work objects assigned to operator update assignment status to pending to verified?

I'll create activity in Assign-worklist class and activity also in that activity and fetch the assignment cases by passing pxAssignedOperatorID=="operator" and I'll fetch the cases of the operator and loop inside assignment class open each and every record and use property-set, objsave and commit, Done.

Q)How do you update individual record?

By using obj-open-by-handle and open the work object by passing the pzInsKey and use the property-set to set values and obj-save, commit, DONE.

Q) In production some work objects updated wrongly how do you achieve?

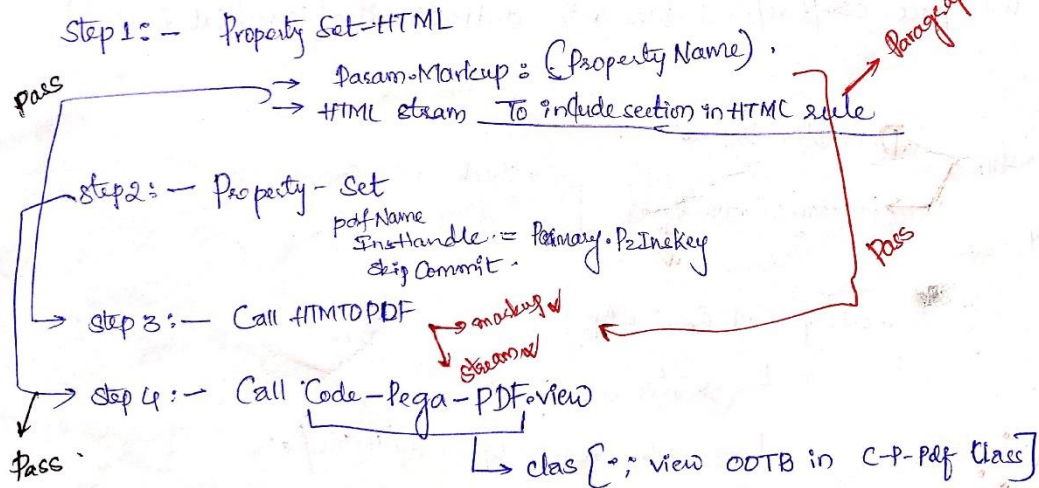
I'll create stand-alone activity and open each and every record by passing pzInsKey, property-set, obj-save, Commit, DONE.

Q) List of records?

Based up on conditions I'll create report and call the report in the activity and I'll open each and every record inside the loop and property-set, obj-save, commit, DONE.

8.1 How to Generate PDF File

PDF
Generate File:



→ Now Get Data from DB to Excel:-

{ ClaimPage.paResults().ClaimID }.

{ ClaimPage.paResults().ClaimHolderName }.

1. Create Template ✓

2. Create Activity

paGenerateExcelFile

Binaryfile (path) → To get template

Give filename (Anything).

Attach file to work obj:-

Step 1-3 same as Generate PDF

Step 4:- Call Code-Pega-PDF.AttachToWork

→ Directed To check

→ [send Email] shape

[Co] [Send] [Call Email With Attachment]

Req 1 → Extract Data From EXL file :-

display

Attach to DB Table

How to Parse Excel file:-

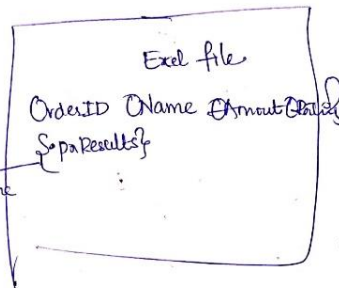
Real Time ✓

{.paResults().OrderID() input}

{.paResults().OrderName input}

→ for Template

name



Excel → Import PageList ✓

1. Create Binary file upload template:

2. In section, add file path control

{ add property } Custom property

filepath

add paRequestor.py file upload in parameters file upload

Pages & Class [add paRequestor Class]

pages

& map to property }

data uploaded

System pages → paRequestor →

3. Create Activity, Call paParseInput

parameters

FSFileName: paRequestor.py file upload → Code-Pega-Requestor
Template: "WebObjUpload.xls"

Any Issues
→ Check Binary file after download ✓

Pages & Classes

Ord Page → Code-Pega-List

paRequestor → Code-Pega-Requestor

Ord Page. Py Sheets().paRequestor → \$

checkboxes ✓

Output PageName:

Now we can see the file in clipboard ✓

4. Loop & Attach to PageList

5. attach data to Table by [from Clipboard]

→ open clipboard page in loop

→ set values to Data-Table map ✓
Obj-Save

paFlow
→ To check subscription
→ paSubscription

9. LISTENERS

9.1 EMAIL Listeners

Email Listener will listen the email from "INBOX" folder and process the email into the pega application with the help of Service Email.

PEGA Application<-----InBound Email-----External Source

To process the Email into the Pega Application we need to create below rules.

1. Create Properties to map the Email Details like To, From, Body of the Email, Subject, CC..etc
2. Create a Service Package rule

We need to Provide the Process mode, Access group, Authentication and Service method (Service Email)

3. Create Email Listener rule

Email Listener will listen the email from "INBOX" folder and process the email into the pega application with the help of Service Email.

In a Email Listener we need to provide the Email Account, Service Package, Service Email, Folder Path.

4. Create Service Email rule

By using Service Email we will process the Email into the pega application with the help of Service Activity.

Service Tab:

➔ In a Service Tab we need to provide the Class Name, Page Name and Activity name

Request Tab

- ➔ In a Request Tab we need to map the "To,CC,From,Subject,Sender name, Email Body...etc properties

Response Tab

- ➔ In a Response Tab we need to map the Success message templates

5. Create Activity

We need to call the "CreateWorkFromMail" Activity Create Case and process the case in pega application.

Ex: pyCreateAndManageWorkFromEmail Pega OOTB activity to create cases in Pega.

9.2 FILE Listeners

File Listener will listen the file from Destination location (Application Server Path) and process the file into Pega application with the help of Service File.

PEGA Application<-----Txt,CSV,EXcel-----External Application/Source

To process the file into pega application we need to create below rules.

1) Create Properties to map the "Txt,CSV,EXcel" Files Data.

2) Create Service Package rule

In a Service Package we will provide the Process Mode (Stateless or Statefull), Access Group and Authentication.

3) Create Service File rule

By using Service File we can process the file into pega application with the help of Parse delimiter rule and activity

4) Create Parse Delimiter rule

By Parse Delimiter rule we can comma separated CSV Data into Clipboard

5) Create Activity

In a Activity we will write a logic to Insert or update or To Create Cases in pega based upon requirement.

6) Create File Listener

File Listener will listen the file from Destination location and process the file into pega application with the help of Service File.

In a File Listener we need to provide the Service Package,Service File,Type of the file to Process and Destination location.

Connect-File:

Connect File will place the "Txt, CSV, EXcel" file into destination location.

PEGA Application-----Txt,CSV,EXcel----->External Application/Source

To place the file into Destination location we need to create below rules.

- Create Properties
- Create Connect File

→Connect File will place the "Txt, CSV, EXcel" file into destination location.

→In Connect File we will specify the Destination path and File Name.

- Create Activity

In a Activity we will call the Connect File rule by using Connect File Method and write a logic to insert the data into EXcel or txt or CSV files.

9.3 Agents

Agent is a internal backgroup Process, Automatically it will run the activity based sometime interval like "Seconds, Min, Hours, Daily, Weekly, Monthly, Yearly and Timezone" bases.

TYPES OF TIME INTERVALS:

Periodic:- If we want to run the Agent Activity based upon "Seconds" then we will select "Periodic"

Recurring:- If we want to run the Agent Activity based upon "Daily, Weekly, Monthly, Yearly and Timezone" then we will select "Recurring"

Startup:- If we want to run the Agent Activity when we are starting the Server then we will select "Start Up".

TYPES OF AGENTS:

1.Standard Agent (Queue Processor):

- If we want to update Work Object related data in a background while creating a Work object through File Listener or Email Listener or Service SOAP or REST.
- If we go for Standard Agents, Pega itself will handle Exceptions and locking Mechanism. For Example If Any one the Work object is failed to process or update then it will go to the Broken Queue. Pega automatically it will requeue the Broken Queue Work Objects and process again.
- Standard Agent will follow the "Auto Queue Management (AQM)" means it will follow "FIFO"

Ex: Queue < 1---2----3-----4-----5

→ By using "QueueForAgent" PEGA OOTB Activity or "Queue-For-Agent" method in a activity we can queue the Work Objects to Standard Agents. We need to pass the "AgentRuleset Name, Agent Name, Number of Attempts to retry and Minimum Age Processing"

→All the Queued Work items will be stored into "pr_Sys_Queues" Data base table.

Note: System-Queue-DefaultEntry class to "pr_Sys_Queues" Data base table

2.Advanced Agent(Job Scheduler)

- Whenever we want to send remainder emails or Invoice Reports or to do the Purging Activity means Delete the Older Work Objects, To Delete the Operators whoever not log in your application more 180 days or TO Delete or Update Staging Data tables related data then we can go for Advanced Agent.
- If we go for Advanced Agent, we need to handle Exceptions and locking Mechanism in activity. In case when we are update the work objects by using Advanced Agents, If any one of the work object is failed to process then we need to process the failed work objects manually.
- Advanced Agents will not follow the "Auto Queue Management (AQM)"

Agent Scheduler:

- Agent scheduler will take care to run the agent activity based upon time intervals.
- Agent Scheduler rule no need to create manually. when we created a agent rule, Pega itself will create the agent scheduler

Note:-Agent Scheduler rule no need to Deploy into other environment's like SIT OR UAT or PROD

Note: if we want to reschedule any agent or we want to run the agent activity Again then we need to update the "Agent Scheduler".

How to Trace the Agent:

IN PEGA 8V

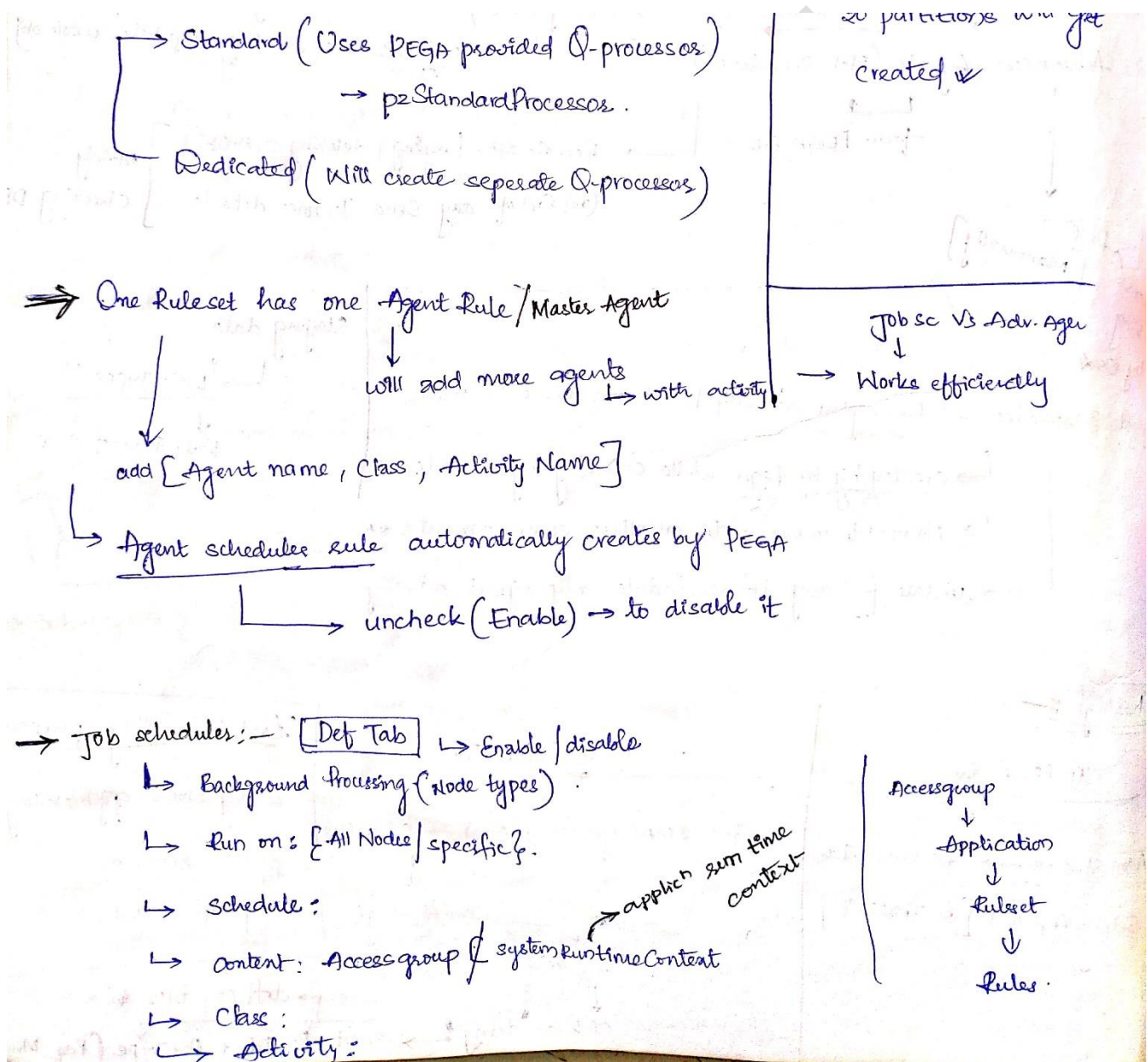
- ➔ Goto the admin Studio and Click Resources
- ➔ Select the Agent, Now we can see the List of agents available in our Application
- ➔ Which agent we want to trace, we need to select that particular agent and click on Trace.

IN PEGA 7V

By using SMA (System Management Application) we can trace the Agent

- ➔ Goto the SMA and Select Node
- ➔ we need to click on "Agent Managenet"
- ➔ Select the Agent, Now we can see the List of agents available in our Application
- ➔ Which agent we want to trace, we need to select that particular agent and click on "Delay"
- ➔ Goto the Request Management and Requestor ID
- ➔ Now click on "Tracer".

Class Name	Table Name
System-Message-QueueProcessor-DelayedItem	→ pr_sys_delayed_queue
System-Message-QueueProcessor-BrokenItem	→ pr_sys_msg_qp_brokenitems
System-Queue-DefaultEntry	→ pr_Sys_Queue
System-Queue-ServiceLevel	→ pr_sys_Queue_sla



9.4 Queue Processor

1. In a Queue Processor, PEGA will use the Kafka Messaging Queue or System to queue the work objects to Queue Processor.
2. Each Kafka Messaging Queue or System will have 20 partitions means at a time it will process 20 work objects.
3. By using Run in BackGround Smart shape in a flow /Queue-For-Processor Method in a actcivity we can call the Queue Processor.

TYPES OF QUEUE PROCESSOR:

1. Standard: → it will uses the pzStandardProcessor pega ootb Queue Processor
2. Dedecated: → we will create Seperate Queue Processor.

Note:

- In a Queue Processor, all Delayed Work Objects will be stored into pr_sys_delayed_queue table.
- In a Queue Processor, all Broken Work Objects will be stored into pr_sys_msg_qp_brokenitems table.
- SLA Related Queue Work objects will stored into pr_sys_Queue_sla table.

pr_Sys_Queue → All Work Objects → One by one FIFO 1Lac

Kafka Messaging Queue or System → 20 partitions → 20 Thread → 20 Work Objects

QueueForAgent PEGA OOTB/Queue-For-Agent method

Run in BackGround Smart shape in a flow /Queue-For-Processor Method in a actcivity

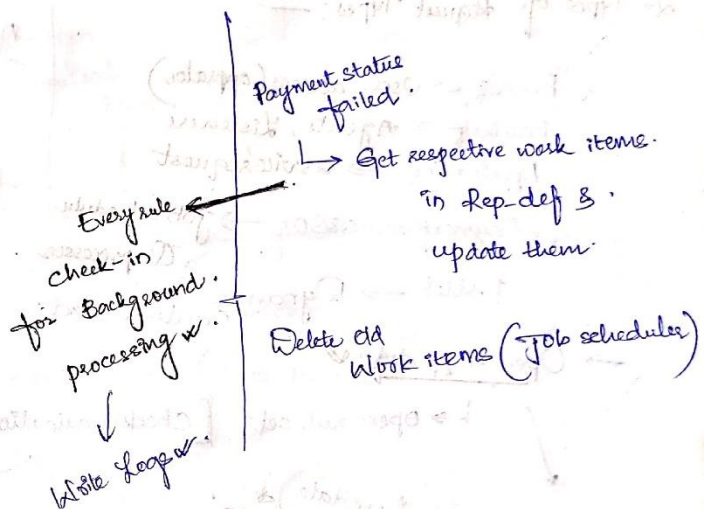
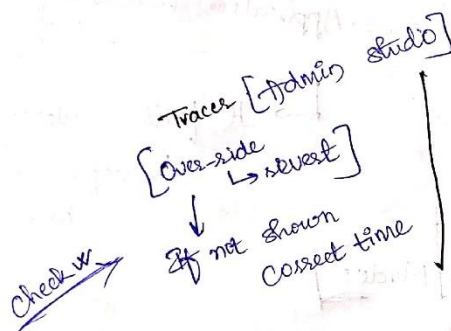
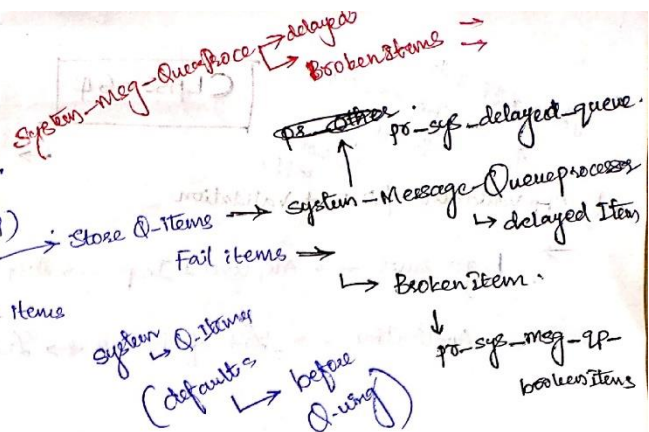
→ Q. Processor :- Definition Tab

- Enable / disable.
- How to run (background processing).
- When to process (Immediate / Delayed)
- No of Threads per Node
3x20 → 60 Work Items

→ Class Name.

→ Activity

→ Processing series :-



CLASS - 62

→ update case status while creating a case { Q-processors }.

Q-processor → by Utility shape run in background.
→ by Activity method.

→ after saving 20 partition gets created
[i: kafka msg system]

Queue-For-Processors
→ removed after successful processing

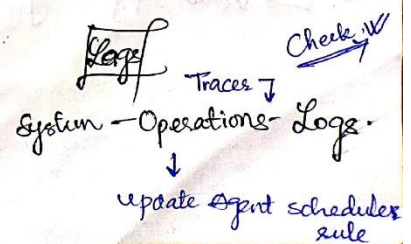
→ Standard (ps Standard Processors)
→ Advanced (He creates separate Q-processors)

→ Agents :- Schedule Tab → Master agent rule.

→ We can add multiple agent rules here.

→ We can disable here.

Q-For-Agent → Use Activity → Pre-defined Activity [we can use Already created Agents]



10. CASE MANAGEMENT

what are the options available in a Stage Level:

1. We can set Case Status while entering into the Stage
2. We can execute Stage based upon when Conditions
3. We can define the Validation rule before entering the Stage
4. We can define the SLA (Time and Urgency) To complete the Particular Stage

what are the options available in a Process Level

1. We can execute the Process based upon when conditions
2. We can define the SLA(Time and Urgency) To complete the Particular Process

what are the options available in a Step Level

1. We can route the Assignment to Users,Work Queues
2. We can define the Work Object Status in a Assignment level
3. We can define the SLA(Time and Urgency) To complete the Particular Assignment

How to Add One More Process to Stage

1. We can the New Process from "pyDefault" Case Type rule under Stages Tab
2. Open Case Type from Cases Explorer → Select the Stage → Click on Dotted symbols → Add Process → New Process

How to Add the Parallel Process to Stage

1. If we want to run the two Processes(Flows) parallelly then we can add the Parallel process from Case Life cycle.
2. Open Case Type from Cases Explorer → Select the Stage → Click on Dotted symbols → Add Parallel Process → New Process

How to Add the Alternative Stage in a Case Life Cycle

1. If we want to Define the Exceptional path or Alternative path to Primary Case Life Cycle then we can Add the Alternative Stages.
2. Open Case Type from Cases Explorer → Select the Stage → Click on Dotted symbols → Add Alternative Stage.

How to Change from one Statge to Another Stage

1. By using "ChangeStage" automation shape we can jump from one stage to another stage.
2. By using "ChangeStage" local action we can jump from one stage to another stage
3. By using "pxChangeStage" PEGA OOTB activity we can jump from one stage to another stage

Ex:Leave Request

Stage1 Leave Details Statge 2 Review Leave Details Stage3: Approval

pyApprovalResult=Approved

How to Add SendEmail Automation shape in a Case Life Cycle

1. By using SendEmail Automation shape we can send to external Source.

How to Add the Create PDF Automation shape in a Case Life Cycle

1. By using "CreatePDF" Automation shape we can create PDF in pega

How Identify the Duplicate Cases in PEGA

1. By using "Search Duplicate Case" Automation shape we can indentify the Duplicaes in pega

Q) What is the use the of "Case match" rule in pega?

By using Case Match rule we can indentify the duplicate cases in pega. In a Case Match rule we will specify the "Must match conditions and Weighted match conditions" to find the duplicate cases.

Ex: Raise Complaint Prodcut Com-3535

Dish Tv

Raised

Bro One more

How to Create a Temporary Cases in PEGA

1. By using "Persist" Automation shape we can create temporary cases in pega

Note:-In case Type under "Settings" tab we have "Create temporary case until r eaches to "Persist shape" check box is available. we need to check

that check box to create temporary cases and include in Persist automation shape in a flow.

How to Create Child Case in pega

1. By Using "CreateCase" automation shape we can create a Child Case

Note:

- when we are in child Case, Child Case data is available in a clipboard under User Pages with page name "pyWorkPage" and parent data is available in a clipboard under pages with page name "pyWorkCover"
- pxCoveredCount → it contains the Child case count
- pxCoveredCountOpen → it contains the "how many child cases in open Status"
- pyWorkPage.pxCoveredInsKeys → When we are in parent case, if we want to get child case id then we can see in clipboard under user pages with page name "pyWorkPage.pxCoveredInsKeys"
- pxCoverInsKey: It contains the Parent Case ID when we are in child case.

Ex1: Claim Application

Vehical V-11

Health

Fire

Ex: OnBoaring Parent Case

Employement Child Case

Case Dependency in PEGA

1. By using "Wait" shape we can achieve the Case Dependency in pega

Note: Data Propagation is a nothing but copying the data from Parent case to child case or Child case to Parent Case

How to Propagate the Data from Parent Case to Child Case

1. In a Parent Case type under "Settings" Tab we have "Data Propagation" option is available, there we can map the parent case and child case properties

How to Propagate the Data from Child Case to Parent Case

When we are child Case, First We will open Parent Case by using Obj-Open-By-Handle Method activity passing "pxCoverInsKey" and set the Child case data to parent case by using Property-set method. Now we need to call the activity while submitting the child case.

- Open Child by using Obj-Open-By-Handle and passing "pxCoverInsKey" value
- Use the Property-Set to set Child case data to parent case
- Call the Activity in child case flow action under post processing

Lock Mechanisam in PEGA

1. In a Case life cycle under "Settings" tab we can specify the Whether it is default locking or Optimistic Locking

2. In a Child Case, Locking strategy is inherited from parent case type
3. If we want to give the access to work on the parent case and child case different users same time then we need to check the "Allow other users to access parent case when the child case is opened" check box in a child case under "settings" tab
4. All the Locking related data will be stored into "pr_Sys_Locks" Table
5. System-Locks class belongs to "pr_Sys_Locks" Table

Default Locking(Allow Single User):-

- ➔ Default Locking means only one user can work on any particular case at a time.
- ➔ Default Locking time out is "30" min

Optimistic Locking (Allow Multiple Users):-

- ➔ Optimistic Locking means multiple users can work on any particular case same time.

How to Acquire the lock on the Work objects in pega?

Obj-Open,Obj-Open-By-Handle,Open-Delete-By-Handle,Obj-Refresh-And-Lock and "acquireWorkObject" pega ootb activity

When it will remove or release the lock on the work object?

when are committing the Work object,pega automatically it will release lock from "pr_Sys_Locks" Table.We need to check the "Release on Commit" check

box under Obj-Open, Obj-Open-By-Handle, Open-Delete-By-Handle and ReleaseLock Pega ootb activity

Work Party in PEGA

1. If we want to associate any interested work party to our case Life cycle then we can define the work Party.
2. Work Party rule can be called in a case Life cycle under Settins tab we have "Participants" option is availble,there we can call the work party roles.
3. By using "addWorkObjectParty" pega ootv activity we can call the Work Party rule.

Note:Work Party classes are inherated from "Data-Party-" class

Where we can see the work party details in a clipboard?

Inside the pyWorkPage we have "pyWorkParty" page group property is available,there we can see the added work party details.

Ex:pyWorkPage.pyWorkParty(Owner)

Ex:pyCaseManagementDefault

Ex:Complaint

First Name Last Name Phone Nume Email ID

Complaint Case Life Cycle

Customer ----->CSR-->Complaint Details--->Technical Team

10.1 Skin

CLASS-72

- Components like:
 - Buttons, Labels, Layouts, Errors etc--
 - Can customise styles----

→ Mining :- {style pattern}.

↳ Style of Text, Border & Background.

↳ Any change for format
create new format [X don't override]
↓
ref it in particular places.
↳ Layout + Formats.

→ don't change existing one:

→ We can refer, CSS Class styles:-

→ In portal Level, we can define skin rule--

{ from Uses Interface → skin, on existing skin (pega's) }

↳ py EndUse

→ One of Rule in "PEGn"

SKIN

↳ Presentation (UI), styling

↳ Component Styles/Tab

↳ Mining Tab

↳ Typography

↳ [Text style]
↳ Background:
↳ Borders.

↳ Combinations.
{above 3}.

↳ My } Inherited
↳ Own } Pega's
↳ Created

Q: Any Idea (Yes)

Q: What to do? If we want to modify.

Q: Minin? Text style, Border, Background

Q: Component style?

↳ Layouts, Buttons, dropdowns etc--

Harness :- Harness

↳ How you want show UI to enduses. [physical appearance of section].

↳ we will call sections in-Harness rule.

1. New

2. Perform

3. Confirm

4. Review

↓
we can
customise
these

Even we have more ✓

→ At the time of creating case (all the fields will be called).

→ Entering data in Properties

→ After submitting data

→ When Case is re-opened ✓

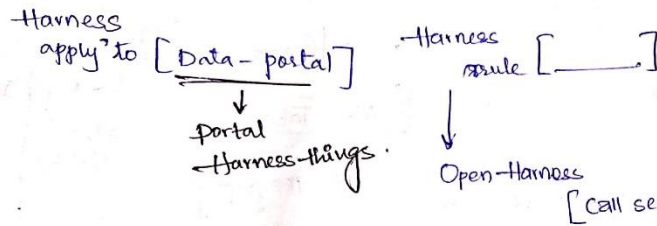
Performed screenflow
Tabbed screenflow

→ These are also harnesses--

→ access under Definition Tab → Access group

→ Postals {

UI → create → Postal after creating



→ To add sections {work-objects} in Harness (or) Data Pages
we need to add respective classname under pages & classes of
section and call it {ref: Clipboard page}.
page context---

WF - Data - Customer

WF - USA - Sale - Bank 42 - Work - Credit

- Complaint
- Claim

property ref by
page property

Pre-defined
Harness rules?

If want section
[by clipboard pagesreference]
also---